AFRL-RX-TY-TM-2009-4554

DEVELOPMENT OF SENSOR COMPONENT FOR TERRAIN EVALUATION AND OBSTACLE DETECTION FOR AN UNMANNED AUTONOMOUS VEHICLE

Sanjay C. Solanki

University of Florida
Mechanical and Aerospace Engineering
231 MAE-A
Gainesville, FL 32611

MAY 2007

A dissertation presented to the Graduate School of the University of Florida in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

AIRBASE TECHNOLOGIES DIVISION
MATERIALS AND MANUFACTURING DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
139 BARNES DRIVE, SUITE 2
TYNDALL AIR FORCE BASE, FL 32403-5323

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory, Materials and Manufacturing Directorate, Airbase Technologies Division, Public Affairs and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

REPORT NUMBER AFRL-RX-TY-TM-2009-4554 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//signature//.
WALTER M. WALTZ
Work Unit Manager

//signature//.
SANDRA R. MEEKER, DR-IV
Acting Chief, Force Protection Branch

//signature//.
ALBERT N. RHODES, PhD
Acting Chief, Airbase Technologies Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 01-MAY-2007	**2. REPORT TYPE** Dissertation	**3. DATES COVERED** *(From - To)* 01-MAY-2006 -- 01-MAY-2007

4. TITLE AND SUBTITLE
Development of Sensor Component for Terrain Evaluation and Obstacle Detection for an Unmanned Autonomous Vehicle

5a. CONTRACT NUMBER
FA4819-05-D-0011-0001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
99999F

6. AUTHOR(S)
Solanki, Sanjay C.

5d. PROJECT NUMBER
GOVT

5e. TASK NUMBER
F0

5f. WORK UNIT NUMBER
QF503008

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Florida
Mechanical and Aerospace Engineering
231 MAE-A
Gainesville, FL 32611

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Materials and Manufacturing Directorate
Airbase Technologies Division
139 Barnes Drive, Suite 2
Tyndall Air Force Base, FL 32403-5323

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RXQF

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RX-TY-TM-2009-4554

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A: Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Ref AFRL/RXQ Public Affairs Case # 09-121. Dissertation submitted to the University of Florida in partial fulfillment of the requirements for the Degree of Doctor of Philosophy. Document contains color images.

14. ABSTRACT

Applications of autonomous vehicles are found in various diversified fields including defense, agriculture, mining, and space exploration. For successful operation of an autonomous vehicle, the vehicle has to dynamically interact with the environment around it. This dissertation presents the development and implementation of a sensor component to represent the surrounding environment around an autonomous vehicle. The environment is modeled as a grid and each cell in the grid is assigned a traversability value. Each of the two algorithms, obstacle detection and terrain evaluation, have certain advantages and limitations. Also like any other sensor algorithm, these two algorithms have some uncertainties associated with their outputs. In my work these algorithms are fused together to complement the uncertainties and limitations of each of them, and outputs of the two algorithms are fused using the certainty factors approach.

15. SUBJECT TERMS
University of Florida, robotics, autonomous, traversability, algorithm, LADAR, sensor, obstacle detection, terrain evaluation, JAUS, obstacle mapping

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Walter M. Waltz
U	U	U	UU	131	**19b. TELEPHONE NUMBER** *(Include area code)* 850-283-3725

Reset

DEVELOPMENT OF SENSOR COMPONENT FOR TERRAIN EVALUATION AND OBSTACLE DETECTION FOR AN UNMANNED AUTONOMOUS VEHICLE

By

SANJAY CHAMPALAL SOLANKI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2007

To my sister, Seema.

ACKNOWLEGEMENTS

I would like to thank my committee members; Dr Carl Crane, Dr Warren Dixon, Dr John Schueller, Dr Antonio Arroyo and Dr Douglas Dankel.

Especially I am very thankful to my advisor, Dr. Carl Crane, who has been a great support and inspiration for the entire course of my graduate studies. I would like to thank David Armstrong, for his support as a project manager. I thank Dr Mike Griffis, for putting all the efforts in building the NaviGator. I express my thanks to all the students at CIMAR including Bob Touchton, Tom Galluzzo, Daniel Kent, Roberto Montane, Jaesang Lee, Shannon Ridgeway, Ji Hyun Yoon, Steve Velat and Greg Garcia, who have accompanied me on this challenging and exciting path. I would like to thank everybody at the Air Force Research Laboratory at Tyndall for their support.

Thanks to my wife Rita, my daughter Risha, my parents, and my sisters who have always been there to help me.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Figure page

LIST OF OBJECTS

Object page

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DEVELOPMENT OF SENSOR COMPONENT FOR TERRAIN EVALUATION AND OBSTACLE DETECTION FOR AN UNMANNED AUTONOMOUS VEHICLE

By

SANJAY CHAMPALAL SOLANKI

May 2007

Chair: Carl Crane III
Cochair: Warren Dixon
Major: Mechanical Engineering

Applications of autonomous vehicles are found in various diversified fields including defense, agriculture, mining, and space exploration. For successful operation of an autonomous vehicle, the vehicle has to dynamically interact with the environment around it. My dissertation presents the development and implementation of a sensor component to represent the surrounding environment around an autonomous vehicle. The environment is modeled as a grid and each cell in the grid is assigned a traversability value. This grid, termed as the Traversability Grid, forms an internal representation of the state of the environment in real time. The Traversability Grid aids the autonomous platform to drive on a good traversable path and avoid non-traversable regions around it.

The real world environment is highly unstructured and dynamic. It is very difficult to discretize the world into two variables: good and bad, hence the sensor component developed in this dissertation uses a scale of traversability values to classify the region around the vehicle. LADAR (Laser Detection and Ranging) sensors are used to sense the environment. The raw data from the LADAR sensors are processed using two different algorithms; obstacle detection and terrain evaluation. The obstacle detection algorithm uses a weighted sum of evidences method to

find if a region is occupied or free. The terrain evaluation algorithm maps the terrain in front of the vehicle. Based on the geometry of the terrain, it assigns a traversability value to the region. Features used to find the traversability include discontinuity in the terrain, slope of the terrain based on the least squares method, and the variance in the point cloud.

Each of the two algorithms, obstacle detection and terrain evaluation, have certain advantages and limitations. Also like any other sensor algorithm, these two algorithms have some uncertainties associated with their outputs. In my work these algorithms are fused together to complement the uncertainties and limitations of each of them, and outputs of the two algorithms are fused using the certainty factors approach.

CHAPTER 1
INTRODUCTION

Motivation

The concept of unmanned autonomous vehicles is one of today's leading research areas. An autonomous vehicle can be defined as a vehicle which would drive by itself with a limited degree of human intervention or possibly no human intervention at all. So far these vehicles have found applications in various fields such as military, planetary exploration, agriculture, and mining. The Defense Advanced Research Projects Agency (DARPA) has taken a lead role in encouraging and pursuing the autonomous vehicle technology. The large participation and excitement associated with the DARPA Grand Challenge National event in 2004 and 2005 is an indication of the importance of autonomous vehicles in today's world.

As shown in Figure 1-1, the operation of an autonomous vehicle can be classified into four main components, of these the perception of the environment and vehicle control components interact with the real world environment, and the world model and intelligence components store the information and make appropriate decisions. Perception of the environment in real-time is critical for the successful operation of an autonomous vehicle and the research in this paper is focused on the perception element. My research is intended towards representing the real world environment in a format that would help the robot find and traverse the best possible terrain and avoid obstacles as it moves towards its goal. The problem of perception can be categorized into two main sections, one is sensing the environment with the use of various sensors and the other is to represent this data in a form the vehicle can use to maneuver.

To scale the difficulty of the problem imagine driving on rough terrain. How many times would one have to ask the question, "which would be the best way to go?" Would it be desirable to pass the small ditch in order to avoid a big boulder or would it be better to traverse the uneven terrain rather than going through the ditch? It is somewhat difficult for humans to answer these questions; how difficult it would be to represent this heterogeneous environment in a digital format for a robot to understand. The problem is not limited to this. The human perception system is very powerful and humans are able to relate to things around them very well; the world is defined in terms of human perception. It is not so easy in the case of sensors. Most of the sensors available today are limited to measure only a limited category of features. For example, one could capture color information with a camera but not depth, so is the same with other commercially available sensors such as laser, infrared, and sonar. Although there are techniques like stereo vision where one can capture both color and depth, they are limited by factors such as environmental lighting conditions and noise. The certainty of any commercially available sensor varies with the environmental conditions. The work in this research is an effort to find an effective solution to the above problems.

Background

Within the Center for Intelligence Machines and Robotics (CIMAR) lab, research in the area of obstacle detection using real-time sensors for autonomous vehicles started with simulations presented by Takao Okui [Okui 99]. Okui presented various techniques for updating local grid maps using simulated sensors. Some of the methods he used included probability theory, Bayes theorem, histogram method, fuzzy logic, and non-homogeneous Markov chains. This work helped the research community in comparing the advantages and disadvantages of each of the above methods. David Novick [Novick 2002] took the research further by incorporating the obstacle detection component on an actual vehicle. He pursued the non-homogeneous Markov chains approach for updating grid maps. He did an excellent job in fusing the output of two non-homogenous sensors, the laser range finder and the sonar sensor.

My work here aims to extend these previous efforts. My objective is to identify traversable regions in an off-road environment that a car-like autonomous vehicle can travel at speeds approaching 30 mph. The next chapter contains a review of the sensors and algorithms already presented by the research community to solve this problem.

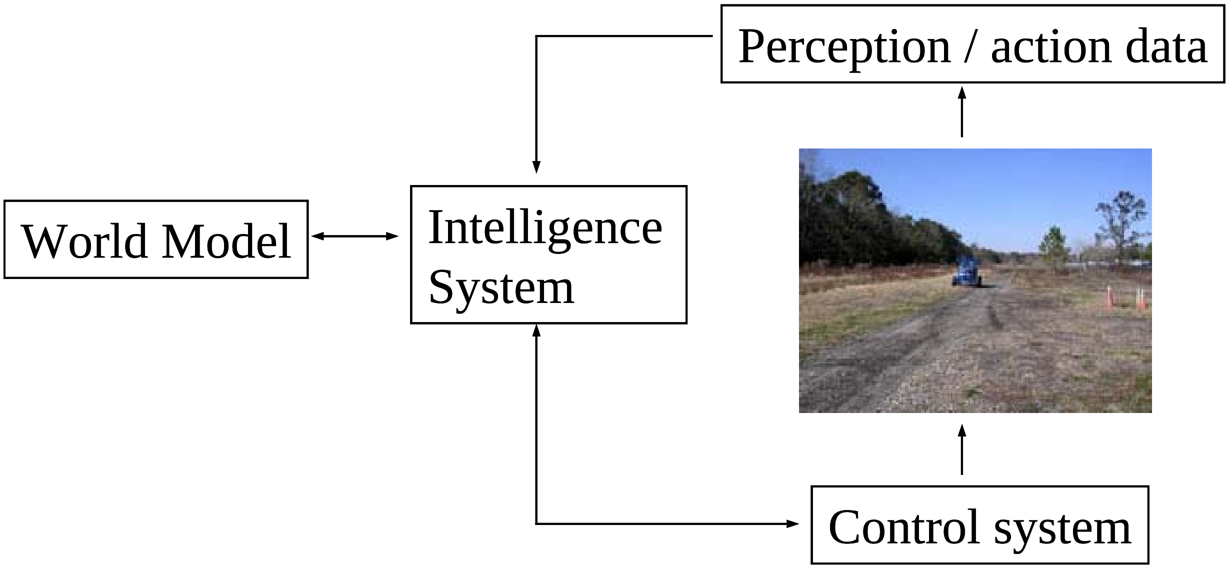


Figure 1-1. Important components of an autonomous vehicle.

CHAPTER 2
LITERATURE REVIEW

Various sensors have been used to date for detecting obstacles and estimating terrain. This chapter begins with a brief discussion on the operating principles of some of these sensors. This is followed by a brief theory on the representation of the environment and managing the uncertainties associated with sensing. The chapter concludes with a review on the literature that has been presented and implemented in the area of real time perception of the environment for autonomous vehicles.

Sensors

Sensors in general can be categorized as active or passive. Active sensors have an internal energy source and emit energy into the environment while passive sensors do not emit energy and depend on their surrounding environment as the source of energy. Both types of sensors are widely used in autonomous vehicle applications. Sensing of the environment includes detecting obstacles and characterizing terrain.

Monocular Vision

Object detection and classification using color has been discussed in the research community for a long time. The primary features used for object detection in this case are the RGB color space. Some of the commonly used algorithms to detect obstacles or classify traversable terrain are edge detection and optical flow. Implementations of monocular vision are found in Davis [1995], Takeda [1996] and Royer [2005].

Stereo Vision

The slightly different perspectives from which our eyes perceive the world lead to different retinal images, with relative displacements of objects (disparities) in the two monocular views of a scene. The size and direction of the disparities of an object serve as a measure of its relative depth; absolute depth-information can be obtained if the geometry of the imaging system is known. Stereo vision can be used to construct a three-dimensional image of the surroundings using the same principle as human eye, provided the geometry and optics of the sensor are well known. An implementation of the stereo vision system can be found in Wu [2002].

Thermal Imaging

Thermal cameras can detect energy at infrared frequencies. All objects emit infrared energy, the hotter the object the more infrared energy it emits. Thermal cameras work better in the night as there exists a better contrast of the surrounding objects. During the day objects absorb heat from the sunlight and tend to blend together. Algorithms such as edge detection are used for classification of obstacles using thermal imaging. Matthies [2003] implements a thermal camera for detecting negative obstacles.

Laser

Most laser sensors use a visible or infrared laser beam to project a spot of light onto a target, whose distance is to be measured. The distance from the spot on the target back to the light-detecting portion of the sensor is then measured in several ways [Carmer 1996]. The general factors to consider when specifying a laser distance sensor include maximum range, sensitivity, target reflectance and specularity, accuracy, resolution, and sample rate. The general methods used to measure distance are optical triangulation and time of flight distance measurement.

The optical triangulation measurement is used to measure distance with accuracy from a few microns to a few millimeters over a range of few millimeters to meters at a rate of 100 to 60,000 times per second. A single point optical triangulation system uses a laser light source, a lens and a linear light sensitive sensor. A light source illuminates a point on an object, an image of this light spot is then formed on the sensor surface, as the object is moved the image moves along the sensor; by measuring the location of the light spot image the distance of the object from the instrument can be determined.

Laser time-of-flight instruments offer very long range distance measurement with a trade off between accuracy and speed. Figure 2-1 shows the two different methods to compute range distance using laser sensors. As shown in Figure 2-1, a short pulse of light is emitted and the delay until its reflection returns is timed very accurately. Since the speed of light is known, the distance to the reflecting object can be calculated and this is referred to as the time of flight measurement. The laser sensor used in this research works on the principle of time of flight. The second method to compute the range distance is by measuring the phase difference between emitted and reflected waves. Example of a laser sensor working on this principle can be found in Hancock [1998].

Ultrasonic Transducers

An ultrasonic transducer works on the principle of time of flight measurement as well. A short acoustic pulse is emitted from the sensor; if the pulse hits an object the return echo is sensed. There are some inherent properties of this sensor which reduce its accuracy of measurement such as, the large wavelength of ultrasonic energy as compared to the surface roughness of an object and its wider beam angle. An implementation of the Ultrasonic sensors can be found in Novick [2002].

Radar

Radar emits electromagnetic radiation. As the signal propagates, objects reflect, refract and absorb the radiation. Large reflecting objects will reflect a stronger signal. The signal strength would be different for different materials. A lower signal strength is received for a large obstacle with high absorptivity. Radars are generally used to detect large metallic obstacles at a distance. Radar based sensor modeling can be found in Foessel-Bunting [1999].

Environment Representation

In order to use the sensors in autonomous robotic applications, the measurements from the sensor are converted into state variables which form an internal representation of the environment. The state variables define the state of the environment at the current instance of time. The two methods generally used to model the environment in autonomous vehicle applications are vector maps and raster maps.

The 2-dimensional vector maps are based on geometric primitives: point, lines, circles, polygons etc. A set of environmental characteristics are tagged with these primitives. These characteristics could be positive or negative obstacles, moving obstacles, obstacle segmentation based on color, etc. Raster maps are built by tessellating the environment around the robot in a 2-D grid. Each cell in the 2-D grid can be classified based on the sensed environment characteristics. The main advantage of a vector map is its compactness. For example only four vertices are needed to define a rectangle of size m by n, while to represent the same rectangle in a 2-D grid map an entire rectangular matrix of cells is required whose size will depend on the size of the rectangle. Generating vector primitives require highly accurate sensor data as compared to generating 2-D raster maps. One of the examples of a raster map implementation found in the literature is the Occupancy Grid map [Thrun et al. 2005]. Occupancy Grids tessellate the surrounding continuous space around the robot into fine grained cells. 2-D Occupancy Grids are more common. In 2-D grids, the information from the 3-D world is represented as a 2-D slice. Each cell in the grid corresponds to a variable, which represents the probability of occupancy of that location.

Uncertainty Management

The sensors discussed above measure a particular characteristic of the environment; hence these sensors provide only partial information of the surroundings. Sensor measurements are often corrupted by noise. Hence there is always some amount of uncertainty associated with the sensor in its representation of the environment. The state estimation algorithms seek to recover the state variables (i.e., the state of the environment from sensor measurements). As it can be inferred, most of these algorithms have to use an uncertainty management mechanism to come up with an estimate of the environment from the sensor readings. Some of these algorithms; probability theory, histogram method, Bayes theorem, fuzzy logic is discussed in Novick [2003].

Thrun et al. [2005] discusses the implementation of Bayes filter, which is a variant of the Bayes theorem used to update Occupancy Grids. The Bayes filter uses the log odds ratio representation instead of the probability representation of occupancy. The log odds representation, l_x is given as:

$$l(x) = \log\left(\frac{p(x)}{1-p(x)}\right) \tag{2.1}$$

where

p(x) is the probability of occupancy.

The log odds ratio is recursively computed as the posterior value from the prior log odds ratio and the current sensor observation as follows:

$$l_t = l_{t-1} + \log\left(\frac{p(x \mid z_t)}{1-p(x \mid z_t)}\right) - l_0 \tag{2.2}$$

where:

l_t is the posterior log odds ratio,

l_{t-1} is the prior log odds ratio,

$p(x \mid z_t)$ is the inverse measurement model which specifies the distribution over the state variable x as a function of the measurement z_t and

l_0 is a constant denoting the prior before any sensor readings.

As seen from the Equation 2.2, the Bayes filter is additive and hence any algorithm that increments and decrements a variable in response to measurements can be interpreted as a Bayes filter in log odds form.

Uncertainty Management in Multi-Sensor Fusion

Any one type of sensor measures a limited characteristic of the environment. Moreover similar sensor hardware used in different sensor algorithms, can obtain different information from the environment. The Bayes filter discussed above is not very useful in combining the outputs from different sensor algorithms. Consider for example, an obstacle which can be detected by one sensor type but cannot be detected by another sensor. These two sensor types would generate conflicting information, and the resulting occupancy grid map would depend on evidences brought by both sensor systems, hence the results will be ill-defined. The information from these multiple sensors is combined by generating a sensor map for each sensor type and then integrate the maps by using a suitable sensor fusion technique. Some of the techniques which can be used to fuse more then one sensor maps into a single map, where each sensor map might provide different information of the environment are discussed below.

Certainty Factors

Certainty factors [Gonzalez 1993] formalism assigns a belief value called as the CF value for a hypothesis, based on the evidence presented to support the hypothesis or evidence presented to contradict the hypothesis. The CF values are expressed as a set of rules having the format:

IF evidence

THEN hypothesis (with a CF value)

A number of such rules are used, each to define the belief in the hypothesis based on the evidence presented. If the evidence supports the hypothesis, the CF value is between 0 and 1. If the evidence contradicts the hypothesis, than the CF value is between 0 and -1. The overall belief in the hypothesis is computed by combining the individual CF values from each rule. The following mathematical relationships are used to combine the CF values:

Let, $CF_{1,}$ CF_2 be the two CF values to be combined. If both the CF values support the hypothesis, the combined CF value is,

$$CF_{combined} = CF_1 + CF_2 \times \left(1 - CF_1\right) \tag{2.3}$$

If the two CF values contradict each other, then the combined CF value is,

$$CF_{combined} = \frac{CF_1 + CF_2}{1 - \min\left(\left|CF_1\right|, \left|CF_2\right|\right)} \tag{2.4}$$

The resulting $CF_{combined}$ is then used as the CF_1 value and combined with another CF value. For the sensor fusion process, consider a number of sensors, which evaluates the environment and presents evidence to either support the presence of an obstacle or supports the presence of a good traversable path (contradiction to the presence of obstacle). The results from these individual sensors can be combined using the certainty factors approach.

Dempster-Shafer Theory

The Dempster Shafer theory [Shafer 1976] was developed through the efforts of Arthur Dempster and his student, Glenn Shafer. It is based on a mathematical theory of evidence where a value between 0 and 1 is assigned to some fact as its degree of support. The theory allows belief values to be associated with sets of facts as well as individual facts.

For example consider a case, where there are two possible conclusions. The set of possible conclusions is, $\Theta = \{\theta_1, \theta_2\}$. The frame of discernment is defined as the set of all possible conclusions. This set is, $\{\phi, \{\theta_1\}, \{\theta_2\}, \{\theta_1, \theta_2\}\}$, where, ϕ, represents an empty set.

A mass probability function assigns a numeric value from the range [0, 1] to every set in the frame of discernment. The mass probabilities are assigned based on the support that the evidence presents for each of the conclusions. The sum of these mass probabilities (i.e., the total probability mass) is 1.

The belief, *Bel* of a subset A is the sum of all the mass probabilities, *m*, assigned to all of the proper subsets B of A:

$$Bel\left(\{\theta_1, \theta_2\}\right) = m\left(\{\theta_1\}\right) + m\left(\{\theta_2\}\right) + m\left(\{\theta_1, \theta_2\}\right) \tag{2.5}$$

The certainty associated with a particular subset A, is defined by the belief interval, $\left[Bel(A) \quad P^*(A)\right]$, where, $Bel(A)$ is the measure of total belief in A and its subsets and $P^*(A) = 1 - Bel\left(A^c\right)$ is a measure of failure to doubt A.

Finally for combining two evidences, the mass probabilities from each of the evidence are combined using Dempster,s combination rule. Consider mass probabilities, *m₁* (representing a preexisting certainty state associated with hypothesis subsets X) and *m₂* (representing some new evidence associated with hypothesis subsets Y), these two are combined into a mass probability *m₃* (representing the certainty state associated with C, the intersection of X and Y) as follows:

$$m_3(c) = \frac{\sum_{X \cap Y = C} m_1(X) \times m_2(Y)}{\sum_{X \cap Y = \phi} m_1(X) \times m_2(Y)} \tag{2.6}$$

Neural Networks

Artificial neural network (ANN) is an information processing model that is inspired by the way biologic nervous systems such as the brain, process information. The ANN model is made of strongly connected groups of simple processing elements called neurons. These neurons are activated when their input is higher than some threshold value. When the neurons fire, they communicate their value to their downstream connected neurons, which perform the same function. The main advantage of ANN’s is their ability to learn from training data. The learning process may be supervised (backpropagation method) or unsupervised learning (Kohonen network). The major limitation of neural networks is the problem of developing a general representation from a limited amount of training data.

Sensor Implementations in Autonomous Vehicles

The previous section introduced the various types of sensors used in autonomous vehicles for real time perception. A brief theory of some of the environment state representation techniques and managing uncertainty associated with sensor results was discussed. In this section some of the methods and algorithms used to implement the sensors are presented.

In most of the methods, terrain mapping is described as building a Cartesian elevation map from range sensor data. However using this method for mobile robots has certain drawbacks, such as the elevation data can be sparse and non-uniform (this is more apparent as the distance from the sensor increases and because of the occlusion of regions due to obstacles.) To deal with the above problems [Kweon 1991], implements a locus method for building 3D occupancy maps using successive range images obtained from a laser range finder. In this method the elevation z, at a point *(x,y)* is found by computing the intersection of the terrain with a hypothesized vertical line at *(x,y)*. The basic idea of the algorithm is to compute this intersection in the image plane.

[Matthies 2003] describes how a thermal camera is used to detect negative obstacles. The change in the temperature profile of the terrain due to uneven heating and cooling is accounted for. Specifically during night time the interior parts of the negative obstacles tend to remain warmer as compared to the surrounding terrain.

RGB space has been effectively used [Davis 1995] to classify terrain as vegetation or non-vegetation. Two classification algorithms, the Fisher linear discriminant (FLD) and the backpropagation neural network, are discussed. Consider an N-Dimensional space, where the input data could belong to one of the two classes A or B. The FLD algorithm is a linear classifier and generates a linear decision surface that maximizes the correct classifications of A and B.

In many cases the robots are required to work in a dynamic environment. A method for detection of moving obstacles [Kyriakopoulos 2003] and estimating the velocity of the obstacle is discussed and implemented. The paper assumes a polyhedral world i.e., the environment around the robot is composed of linear segments. A single 2-D laser range sensor scans the environment. The range measurements from the sensor are expressed in a world coordinate frame after accounting for the vehicle position and orientation as follows:

$$\bar{x} = \bar{x}_R + r * \bar{e}\left(\theta_r + \varphi\right) \qquad (2.7)$$

where

$\bar{x}$ represents the Cartesian coordinates of the point in the world coordinate system,

$\left(\bar{x}_R, \theta_r\right)$ is the position and orientation of the vehicle,

φ is the angle of the laser ray in the sensor coordinate system and

$\bar{e}\left(\theta_r + \varphi\right)$ represents the unit vector in the direction of the laser ray

Each line segment of an obstacle can be defined by a parametric model as,

$$x = x_0^i + u^i e^i \qquad (2.8)$$

where

x_0^i is a point on the segment and

e^i is the associated unit vector parallel to the segment *i*.

From the Equations 2.7 and 2.8, the intersections of the line segments with the laser beam can be computed. Additional constraints are considered such as the case where the laser ray intersects more than one obstacle line segment for which the range value will correspond to the nearest obstacle. The candidate moving obstacles are then selected on the basis of the obstacle line segments obtained from the successive range reading sets.

Map building algorithms have been developed [Huber 1999] using range sensors. The authors discuss the face-based spin image algorithm to build a map from two laser range scanners, each mounted on a different vehicle.

Kelly and Stentz [1998] developed an adaptive perception algorithm for off-road autonomous navigation. The input to the algorithm is through range images from laser rangefinders and stereo vision. The data from the sensors is processed only in a region of interest that contains the most useful information. The 3D data of the terrain is stored in a 2-D ring-buffer that accommodates vehicle motion through modulo arithmetic indexing. The terrain map discussed eliminates the need of copying the terrain data as the vehicle moves.

Wolf [2005] presents an online algorithm that builds a 3-D map of the terrain using 2-D laser range finder data. The terrain is classified as navigable or non-navigable using Hidden Markov models.

A radar sensor model [Foessel-Bunting 1999] is developed to build three-dimensional evidence grid maps. The grid is updated using the logarithmic representation of Bayes rule.

A perception module which processes range images to identify non-traversable regions of the terrain is developed [Langer D et al. 1994] for autonomous off-road navigation. The sensor used is the ERIM laser which acquires 64 by 256 range images at 2 Hz. The data from the laser range images are converted into Cartesian coordinates local to the vehicle. These data points are then stored in a 2 dimensional grid map. Each cell of the grid map is evaluated for its traversability. The features used to evaluate traversability are the height variation of the terrain within the cell, the slope of the patch of terrain contained in the cell and the presence of discontinuity of elevation in the cell. The algorithm only computes untraversable cells and renders no opinion on smooth traversable terrain.

Bhatavia [2002] proposed a method to detect obstacles on an outdoor terrain which is highly curved with natural rise and fall, but is assumed to be locally smooth. A cost-effective laser scanner system was developed to replace the commercially available high cost two axis lasers. A single axis laser scanner is mounted at an angle of 90° so as to scan a vertical line. The laser is provided with a rotary motion about the second axis to sweep from left to right and register a set of vertical line scans.

The data registered from each vertical scan are classified as an obstacle or free space using a gradient based algorithm. Each of this single scan classification is stored in time-history. The nearest neighbor fusion algorithm generates the candidate obstacles by clustering the group of scans collected in a time window. Experimental results demonstrate that the algorithm is capable of detecting obstacles of size 15 cm in a hilly terrain with minimal false positives; however the terrain has to be very smooth.

The work presented by Talukder [2002a] implements a different approach to classify obstacles based on 3-D geometry for an off-road environment. A conical region is defined around each point in 3-D space. If there is any point in the truncated portion of the cone, both the base point of the cone and the point under consideration are defined as an obstacle. This essentially means for a point to be an obstacle the height difference between the base point which represents the cone and the point under consideration should be greater than a threshold value and the angle between the line formed by the two points and the horizontal surface should also be greater than a threshold value. The 3D points are projected on an image plane and an algorithm to implement the above strategy in the image plane is presented. In the above method, obstacles are represented as pairs of points and this feature is further extended to segment obstacles. All the pairs of points which represent obstacles are defined as linked nodes of an undirected graph. Thus points connected in the chain will represent the same obstacle.

Extension of the above work is presented in Talukder [2002b]. Along with the geometrical properties, color and texture are used to classify obstacles. The author discusses the concept of obstacle negotiation by presenting techniques to predict the dynamic vehicle response to various natural obstacles. The obstacles in a natural terrain such as green vegetation, dry vegetation, hard rock and soil are composed of widely different material properties and yet can be of the same size. Using the color and texture features, the obstacles can be classified based on their material properties. The compressibility of the obstacles is modeled using a spring-damper system. The vehicle is also modeled using a mass-spring system for each of its wheels. The vertical acceleration of the vehicle is computed as it traverses the evaluated path using the obstacle and vehicle models. This predicted acceleration profile is used for obstacle negotiation.

A novel filtering method for terrain mapping, called the Certainty Assisted Spatial (CAS) filter is proposed [Ye 2003]. In this method an elevation map and a certainty map are built from a forward looking 2-D laser range finder. The above filter is compared with the conventional available filters for effective filtering of erroneous measurements due to mixed pixels, missing data, artifacts and noise.

An approach for traversability analysis of rough terrain [Ye 2004] utilizing an elevation map is proposed. The elevation map is defined as E = $\{Z_{i,j}\}$where i and j are the row and column of the cell. A cluster of neighboring cells with non-zero elevation values is defined as an elevation-obstacle. A traversability map is created along with the elevation map to define the traversability index. The traversability index for each cell is evaluated by computing the slope and roughness of a square terrain patch with the cell under consideration as the center of the patch. The size of the patch is equivalent to the number of cells required to accommodate the vehicle in any orientation. Depending on the value of the traversability index, the originally classified elevation-obstacles are re-classified.

In contrast to the general approach of converting the disparity image from stereovision into 3-D Cartesian coordinates to detect obstacles/traversable terrain, [Wu 2002] proposes to use the raw disparity data for 3-D obstacle detection using stereovision. Two different maps are generated: the obstacle map and the slope map. The obstacle map is 75 by 75 elements with each element representing a 0.2 m by 0.2 m area and the slope map is 15 by 15 elements with each element of size 1 m by 1 m. The obstacle map element is marked as one of four possible values; undefined, traversable, negative obstacle or positive obstacle. The obstacle detection algorithm is based on the horizontal isodisparity profiles generated by the stereo image at fixed intervals and the reference lines for each of the isodisparity profiles.

To represent and classify the environment around the robot, a height map is built from range data and fused with the video image [Asada 1988]. The height map obtained from the range image is segmented into four regions: unexplored, occluded, traversable and obstacle. The region where the height information is not available and falls outside the sensor view is tagged as unexplored and the remainder of that region is marked as occluded. The points close to the assumed ground plane and the neighboring points with low slope and curvature are classified as traversable. The remaining regions are labeled as obstacles. The obstacle region is further classified as natural or artificial using the height map and intensity image. The paper assumes that artificial obstacles such as cars have planar surfaces which yield constant slope and low curvature in the height map and linear features in the intensity image while natural obstacles such as trees have fine structures and yield high curvatures in the height map and large variance in the intensity image. A physical simulator of the scale 87:1 was built to perform the experiments.

Most of the commercially available laser range sensors provide two kinds of information for each laser beam reading: the distance of the object being scanned and the intensity of the returned signal. [Hancock 1998] tries to exploit the second piece of information to detect obstacles in a highway environment. Obstacle detection based on range readings as discussed in most of the literature involve building 3-D Cartesian maps and hence involves large computation, not making it suitable for high speed environments. Intensity based obstacle detection for a forward looking laser works on the simple principle that vertical surfaces in front of the vehicle return a stronger signal as compared to the signal returned by horizontal surfaces. This is especially true for larger look ahead distances of the order of 60m where the tilt angle of the laser is very small (about 1° in the present paper). However the major drawback with the above method is that the intensity varies significantly with the surface properties of the object.

Most autonomous vehicle development projects use off the shelf laser range sensors which are not specifically developed for the application. With the advancement in autonomous vehicle technology, commercially available sensors do not meet the specific requirements of range and angular resolution for evaluating wide range of obstacles and terrain at high speed. Carnegie Mellon University has developed a high speed, high accuracy LADAR [Hancock 1998] in joint collaboration with K2T, Inc., and Zoller + Froehlich. The device consists of a two-axis mechanical scanning system with a single point laser measurement system. The scanner provides an unobstructed 360° horizontal field of view and a 70° vertical field of view. The resolution of the system is variable with a maximum of 0.6° per pixel.

Multiple sensors have been used [Stentz 2003] to detect and avoid obstacles in natural terrain. The perception system is categorized in two sensing modules. The appearance sensing module includes color cameras and Forward Looking Infrared (FLIR) cameras, while the geometric sensing module includes LADAR sensors and stereo vision. The geometric sensing module detects geometric hazards from the range data and assigns traversability costs. The traversability costs are further modulated by the compressibility of the sensed terrain. The compressibility is assessed using the range texture, color and color texture to discriminate rigid rocks from bushes.

Sensor data from color and IR cameras are combined with range data from a laser sensor in [Dima 2003] and [Dima 2004] for obstacle detection in an off-road environment. The image space is tessellated into a grid of rectangular cells. The range data are projected into the tessellated grid. Features such as the mean and standard deviation of pixel color in each cell, mean and standard deviation of infrared pixel values in each cell, texture information and the range reading statistics are used as input to the classifier. Instead of combining all the features into one vector and using only a simple low level data fusion, methods to use a pool of classifiers on the subsets of these features and then represent the final output as a fusion of these classifiers are discussed. The classifier fusion algorithms presented in the paper are Committees of Expert, Stacked Generalization and AdaBoost with classifier selection. One of the limitations with the classifier fusion algorithms is the need of supervised learning and hence a large set of training data.

Range imaging sensors are widely used in detecting obstacles with a variety of obstacle detection algorithms. [Matthies 1994] proposes a model to evaluate the different obstacle detection methods. The proposed method for evaluation is divided into two levels: the quality of the raw range data from the sensors and the quality of the obstacle detection algorithm. The quality of the range data is evaluated based on statistical performance models where both the random errors caused by noise and the systematic errors due to artifacts are considered. To evaluate the quality of the obstacle detection algorithm, the model predicts the quality of detecting obstacles and the probability of false positives as a function of the size and distance of the obstacle, the resolution of the sensor and the level of noise in the range data.

[Murphy 1998] discusses the evidential reasoning techniques to fuse sensor outputs. Dempster-Shafer (DS) theory is implemented for sensor fusion at symbol level in the Sensor Fusion Effects (SFX) architecture. The SFX consists of three distinct activities: configuration, uncertainty management and exception handling. The uncertainty management activity collects observations from the sensor and computes the total belief in the percept in three steps. First the observations from each sensor are collected and fused at the pixel level in the collection step. The features extracted from these observations are fused at the feature level in the preprocessing step resulting in a belief function over each sensor’s unique frame of discernment. Finally the fusion step combines these belief functions from different sensors into a total belief in the percept. In the case of a sensor anomaly in any of the steps, exception handling is triggered. Two components from the DS theory, the weight of conflict metric and the enlargement of the frame of discernment, are used for the sensor fusion. The implementation of the above architecture is demonstrated on a mobile robot which acts as a security guard. The outputs from a color camera, a black and white camera, an infrared camera and an ultrasonic sensor are fused to guard a scene.

In an application different from autonomous vehicles, a sensor fusion architecture is designed [Wu 2002] for context sensing. The authors have pointed out the difficulty that lies with fusing different sensors due to the overlaps and conflicts in the outputs and the variations in a sensor's performance with a change in situation. In the experimental example presented, a user's focus of attention is tracked using multiple evidences from different sensors. The output from each of the sensors, a camera and a microphone are viewed as evidences. Two different methods, weighted probability linear combination and Dempster-Shafer theory of evidence, are used to fuse the evidences obtained from the sensors. Since the confidence level of the sensors varies with situation, the authors propose a new concept for a weighted Dempster-Shafer evidence combining rule. The idea is that if we know how a sensor performs historically in similar situations, we can use this information to decide how much confidence we have on the sensors current estimation. The results from the combined estimations are compared against the results obtained using only a single sensor and found to be better in most of the cases.

Rather than extracting sensor data to form an intermediate representation which can be utilized in the planning, [Goodridge 1994] proposes a non-representation based sensor fusion scheme to control robots. A mapping between the sensor data and control signals is developed using Fuzzy sets. The PCFUZ fuzzy development system is developed and used on an indoor mobile robot to conduct experiments involving goal seeking, obstacle avoidance, barrier following and object docking behaviors. The sensors on board include sonar, vision and tactile sensors. Fuzzy rules are created using the outputs from these sensors to produce a continuous surface which defines the control signals. Different techniques such as summation, weighted averaging, fuzzy multiplexing and hierarchical switching are discussed to implement multiple behaviors such as the combination of obstacle avoidance and goal seeking.

Urmson [2006], Miller [2006], Trepagnier [2006] and Braid [2006] presents some of the sensor implementations in the 2006 DARPA Grand Challenge competition. A brief discussion of these sensor implementation is presented below.

Urmson [2006] implements a set of five laser sensors and one radar sensor to detect obstacles and to classify terrain for off-road navigation. Three of these lasers are used to evaluate the terrain while the other two lasers and the radar detect obvious positive obstacles. Each of the sensor processing algorithms generates a cost map. The cost map is a 2-D grid, where each cell in the grid is associated with a cost value which represents the traversability of that cell. The sensor fusion algorithm generates a composite map using a weighted average of each of the sensor cost maps. The terrain evaluation algorithm operates on a single line scan of the laser data instead of a point cloud formed by the successive line scans as the vehicle moves. This approach reduces the effect of imperfect pose estimation. The terrain evaluation algorithm operates by fitting a line to the vertical planar projection of points in vehicle width segments. The traversability cost is computed as a weighted maximum of the slope and the line fit residual.

A grid based terrain representation method, where each cell in the grid, holds not only the elevation values of the cell but also the uncertainties associated with the elevation value is presented in [Miller 2006]. A set of three laser range sensors are used to scan the ground in front of the vehicle. Two of these sensors are fixed on the vehicle, while one sensor is mounted on a two axis gimbaled platform. The gimbaled platform allows the laser to be pointed in any desired direction. The gimbaled laser is combined in a feedback loop with the path planner to gather data along potential paths. The range data from all the three sensors is fused into one map. The probability of the association of each range reading to a particular cell is computed. The probabilistic model takes into account the measurement errors in the lasers itself and the error in the positioning system. The laser measurements assigned to a particular cell are converted into an estimate of the elevation distribution within that cell. The information is stored as an estimate of the cell’s elevation and the associated conditional covariance. Each new measurement is fused with the previous measurement readings, hence there is no need to store the individual sensor measurement readings and only the posterior probabilities of the estimated elevation and variance are stored.

A set of two Sick LMS 291 laser scanners [Trepagnier 2006] are mounted on each side of the front end of the vehicle. Each of these lasers scan the surrounding in a vertical plane and are mounted on an oscillatory platform to provide a 30° oscillating angle. Range data from both the lasers are fused into a common elevation grid map. The data is time stamped and hence obstacles are kept in memory only for a certain amount of time. If the obstacles no longer exist after a specified time, they are removed from the map. This helps in clearing moving obstacles, once they have passed.

Team TerraMax [Braid 2006] selected an array of sensors including a forward looking vision system, single-plane LADAR and multi-plane LADAR. The vision system is based on multi stereoscopic vision and consists of three identical cameras mounted on a rigid bar in front of the vehicle. The three cameras form three different baselines between them. Depending on the speed of the vehicle one of the three camera pairs is selected. The large baseline is used for higher speeds to obtain a deeper field of view; the medium baseline is used for medium speed and the shorter baseline for slower speeds. The multi-plane LADAR is an IBEO ALASCA 4-plane scanner that is used for positive obstacle detection. Two of the planes scan towards the ground and the other two scan towards the sky. The obstacle detection algorithm is based on detecting the slope of the terrain. Obstacles are reported in terms of the closeness of the object collision to the proposed path.

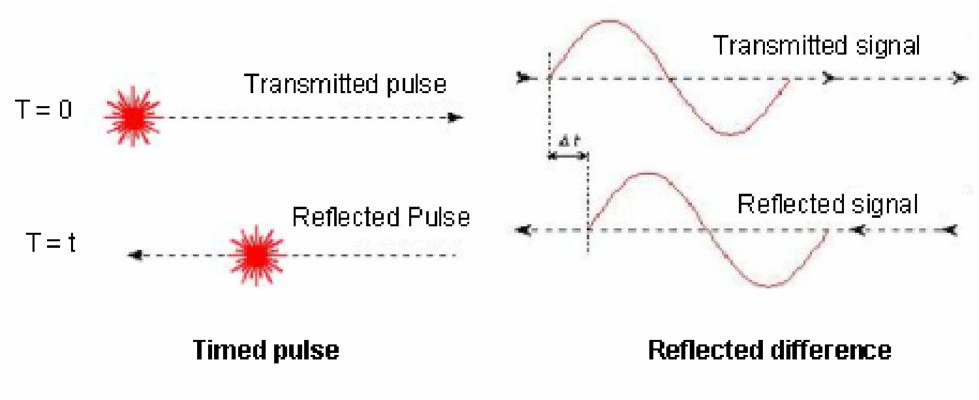


Figure 2-1. Time of flight measurement.

CHAPTER 3
RESEARCH GOALS

Statement of Problem

Given a vehicle capable of maneuvering autonomously and which can obtain its position and orientation measured with respect to a Global Coordinate system, develop a sensor component that will evaluate the terrain and detect obstacles to support autonomous navigation. The sensor component has to be developed for a full-size autonomous vehicle which can traverse on paved and off-road conditions at maximum speed approaching 30 mph. The information should be presented in such a way that the vehicle can use it to avoid obstacles and rough terrain and to seek out smooth terrain.

Research Requirements

Assuming the availability of an autonomous vehicle platform and a positioning system, the steps that are required to be fulfilled to develop a sensor component, that meet the goals mentioned in the problem statement are as follows:

1. Selection of appropriate sensor hardware, which is able to give the required information of the surrounding environment.
2. Selection of proper computer resources to interface the sensor hardware.
3. Design of an environment representation model. The model should not only be able to distinguish the region around the vehicle into traversable and non-traversable areas but should also be able to represent a degree of traversability or non-traversability.
4. Development of algorithms to process the sensor data to detect obstacles and evaluate terrain. Combine the individual algorithms, into one fused output. The fusion process should be able to manage the uncertainties and the limitations associated with the outputs of the individual algorithms. The output of the environment should be in the above mentioned environment representation model.
5. The sensor component should be developed as a Joint Architecture for Unmanned Systems (JAUS) component.
6. Conduct experiments on an autonomous vehicle platform to validate the results of the developed sensor component.

The following sections discuss the autonomous platform and the positioning system used to develop and implement the sensor component.

Autonomous Platform

The development and experimentation work for the sensor component research presented in this dissertation has been done on the autonomous platform, “NaviGator” (Figure 3-1) developed in the Center for Intelligent Machines and Robotics Laboratory at the University of Florida. The platform was initially developed to participate in the 2005 DARPA Grand Challenge competition. Some of the important specifications of the platform are discussed.

Mechanical Specifications

The NaviGATOR’s base platform is a custom built all terrain vehicle. The frame is made of mild steel roll bar with an open design. It has 9" Currie axles, Bilstein Shocks, hydraulic steering, and front and rear disk brakes with an emergency brake to the rear. It has a 150 HP Transverse Honda engine/transaxle mounted longitudinally, with locked transaxle that drives front and rear Detroit Locker differentials (4 wheel drive). The vehicle was chosen for its versatility, mobility, openness, and ease of development.

Power System

The power system consists of two independent 140A, 28V alternator systems. Each alternator drives a 2400W continuous, 4800W peak inverter and is backed up by 4 deep cell batteries. Each alternator feeds one of two automatic transfer switches (ATS). The output of one ATS drives the computers and electronics while the other drives the actuators and a 3/4 Ton (approx. 1kW cooling) air conditioner. Should either alternator/battery system fail the entire load automatically switches to the other alternator/battery system. Total system power requirement is approximately 2200W, so the power system is totally redundant.

Computing Resources

All the computing systems and electronics are housed in a NEMA 4 enclosure mounted in the rear of the vehicle as shown in Figure 3-2. The computing system consists of single processor computing nodes. The system uses a total of eight computers, each of them equipped with an AMD 2 GHz processor and 512 MB RAM. Each node is targeted to perform a specific function of the autonomous system. The system architecture discussed in detail in Chapter 5, explains the breakdown of the autonomous system functionality into these individual nodes. The sensor component developed for this research resides on one computer.

Localization

The NaviGATOR determines its current location using a combination of GPS and inertial navigation system sensor data. The processing and fusing of the navigation data is done by an Inertial Navigation System from Smith's Aerospace. This system is named the North Finding Module (NFM). The module maintains Kalman Filter estimates of the vehicle's global position and orientation, as well as linear and angular velocities. It fuses internal accelerometer and gyroscope data with data from an external NMEA GPS and external odometer. The GPS signal provided to the NFM comes from one of the two onboard GPS systems. These include a NavCom Technologies Starfire 2050 and a Garmin WAAS Enabled GPS 16. An onboard computer simultaneously parses data from the two GPS units and routes the best-determined signal to the NFM. This is done to maintain valid information to the NFM at times when only one sensor is tracking GPS satellites. During valid tracking, the precision of the NavCom data is better than the Garmin, and thus the system is biased to always use the NavCom when possible.

Contributions of this Research

The previous sections described the autonomous platform and the available computing resources to be used for this research. The system explained above is the base foundation and provides all the necessary resources for successful implementation of the research discussed in this dissertation. The contributions of this research can be summarized in short as follows:

1. Development and implementation of an obstacle detection sensor system.
2. Development and implementation of a terrain evaluation sensor system.
3. Development and experimental evaluation of a new sensor fusion algorithm that combine the information from the obstacle detection and terrain evaluation sensor systems.
4. Development of generalized results that determine an optimal sensor system design based on specific vehicle parameters.

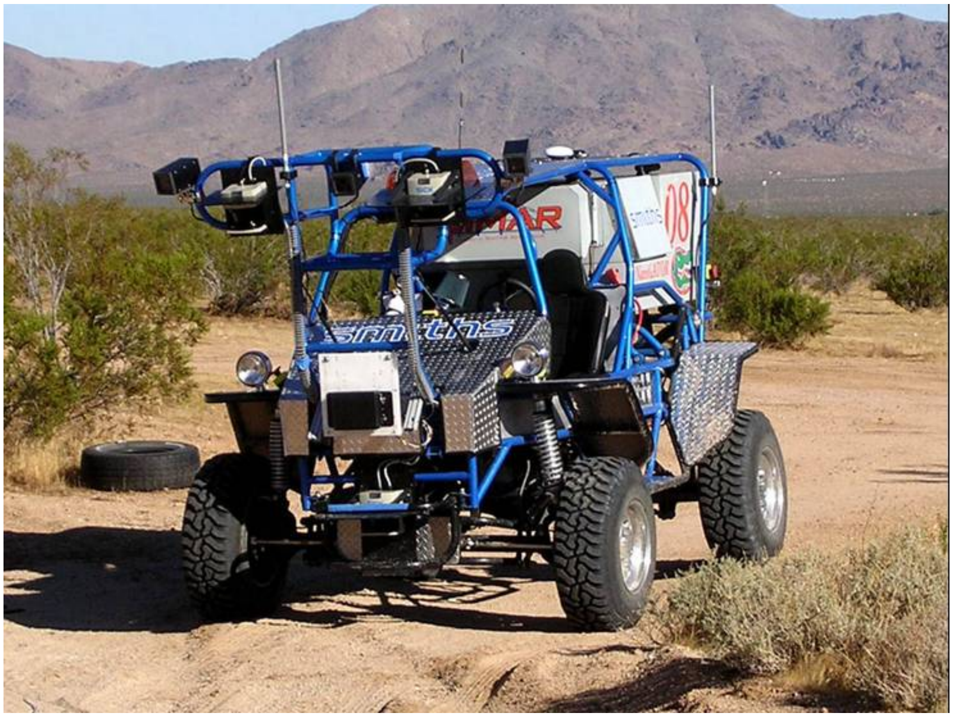


Figure 3-1. Testing platform NaviGator

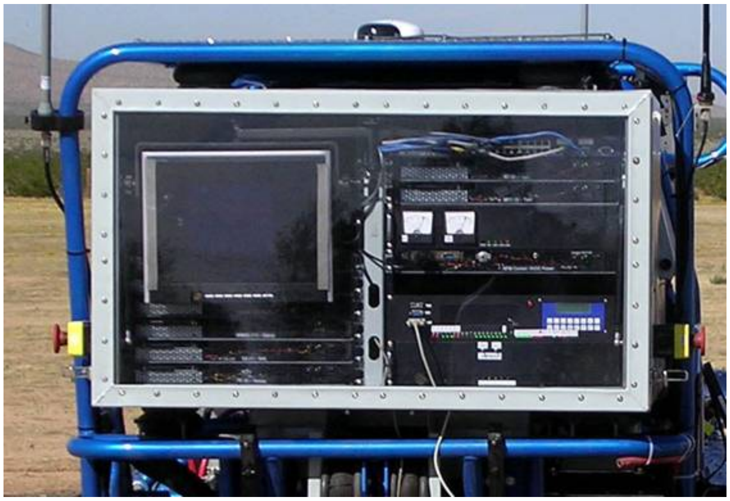


Figure 3-2. Computing resources.

CHAPTER 4
TRAVERSABILITY GRID APPROACH

The surrounding environment of an outdoor autonomous vehicle is highly unstructured and dynamic. The environment consists of natural obstacles which include positive obstacles such as rocks, trees etc. and negative obstacles such as cliffs, ditches etc. There is a possibility of numerous man-made obstacles such as buildings, fence posts, and other vehicles and as well as many other type of objects. The terrain characteristics are also very important for safe navigation. The robotic vehicle should avoid rough, uneven terrain as far as possible and at the same time seek out smooth traversable regions. The representation model of the environment should not only be able to distinguish clearly defined highly non traversable obstacles but should also be able to represent the small differences in the terrain which would for example distinguish a clearly defined paved or smooth dirt path from the surrounding region. This would help to keep the vehicle on the road and at the same time avoid obstacles.

In Chapter 2, two broad classifications of environment representation techniques were discussed; the vector based and the grid or raster based. The current research presents the concept of the Traversability Grid which is a grid based representation of the environment.

Traversability Grid Representation

The Traversability Grid representation tessellates the region around the vehicle into a 2D grid. Figure 4-1 shows the important parameters used to define the Traversability Grid. The grid is always oriented in the North-East direction with the vehicle position in the center of the grid. The grid is defined by the number of rows, number of columns and the resolution of the grid. It consists of an odd number of rows and columns to create a center cell for the vehicle. In the current implementation, the Traversability Grid is 121 rows by 121 columns with a half meter by half meter resolution. This allows a sensor to report data at least 30 m ahead of the vehicle.

Each cell in the grid corresponds to a small region of the environment and is represented by a traversability value. A traversability value of a cell defines the measure of traversability of that cell. The range of traversability value that can be assigned to a cell is from 0 to 15. A real-time sensor component can assign a traversability value between 2 and 12 based on the sensed environmental characteristics. A traversability value of 7 is considered to be neutral; a value above 7 represents a traversable region while a value below 7 represents a non traversable region. As the region becomes more favorable to be traversable, the traversability value of that cell gradually increases from 7 to 12. For example, consider a paved path surrounded by a flat terrain which is not as smooth as the path. Since the flat terrain surrounding the paved path is still traversable, it will be represented by a traversability value above 7; however, the paved path would be represented by a traversability value higher then the surrounding flat terrain, thus although the vehicle could traverse on the surrounding flat terrain, it would be more favorable to stay on the paved road. Similarly, as the region starts becoming non-traversable, the traversability value of the corresponding cell in the grid gradually decreases from 7 to 2. In case the vehicle has to choose to drive through a non traversable region, it would choose to drive in the grid cells whose traversability value is larger. The sensor component outputs a traversability value based on two factors:

1. The severity of a non-traversable region (size of the obstacle or roughness of the terrain) or the good terrain characteristics of a traversable region (smoothness of the terrain).
2. The confidence on the evaluated characteristic. The obstacle might be highly non-traversable but what is the confidence on the presence of the obstacle.

A value of 14 is assigned to a cell whose traversability value cannot be determined by the sensor. The remaining values are reserved for a specific purpose; the value of 0 represents out of bounds, 1 implies to use the same value as last time, 13 is reserved for failure or error and 15 denotes the vehicle location.

Traversability Grid Implementation

The sensor algorithms, discussed in the next chapter, implement the Traversability Grid as a dynamically allocated circular buffer. The circular buffer implementation of the Traversability Grid is very efficient in updating the grid to the new position as the vehicle moves. The main advantage of using a circular buffer in place of a 2-dimensional array is that for a 2-D array, as the vehicle moves, for every new position of the grid, data from the cells in the old grid is copied into the corresponding cell indices in the new grid, this expensive computing operation of copying the grid data every time can be avoided by using a circular buffer. Figure 4-2, shows the change in the grid position due to the movement of the vehicle. As shown in the figure, all the data corresponding to the overlapping cells in the two grids has to be copied from the old grid position to the new grid position for a 2-D array representation.

The circular buffer implementation of the Traversability Grid avoids this computationally expensive operation. The circular buffer stores the data in the grid as a 1D array of size equal to the number of cells in the grid. The position of any cell corresponding to (*row*, *column*) in the 2D grid is mapped into the 1D circular buffer as follows,

$$arrayPosition = \left(row \times numberOfColumns \right) + column \qquad (4.1)$$

When the grid shifts to a new position, instead of copying data from individual cells, the circular buffer defines a pointer to a cell in the grid, which keeps track of the cell corresponding to (0, 0) in the new grid position.

The position of this pointer is defined by two variables, ‘*rowBegin*’ and ‘*columnBegin*’. Initially these variables are set to (0, 0). When the grid moves to a new position, the variables are updated as shown in Table 4-1.

The updated (*rowBegin*, *columnBegin*) are the indices of a cell in the old grid, which correspond to the indices (0, 0) in the new grid. Thus each time the grid moves to a new position the above algorithm recursively keeps track of the start cell of the grid and physically all the data remains in the same array location. The only other operation needed is to clear the data in the cells which no longer overlap.

Since the actual (0, 0) position of the 2-D grid now corresponds to the (*rowBegin*, *columnBegin*), a cell given by the indices (*row*, *column*) is accessed using the algorithm shown in Table 4-2.

The above algorithms define the basic functioning of the circular buffer data structure. Any number of variables of any data type can be stored in the grid using the circular buffer implementation. Each of these variables would be stored as a 1-D array. At the minimum at least one such 1-D array is defined by a sensor component for storing the traversability values of the cells. This array is defined of type, unsigned char. Other arrays may be defined to store information about the cells which could be used internal to the sensor algorithm.

Traversability Grid Propagation

The Traversability Grid data structure is a common data structure representing the environment among all the components of the autonomous platform. Figure 4-4 shows the schematic diagram of the propagation of the Traversability Grid from the Smart Sensors to the Reactive Driver. Each of the Smart Sensor component outputs its own Traversability Grid. These grids are fused into a single Traversability Grid in the Smart Arbiter component. The Reactive Driver uses the Traversability Grid obtained from the arbiter to dynamically compute the vehicle speed and heading and accordingly alters its command to the Primitive Driver, while doing so the Reactive Driver accounts for the traversability value of each cell in the grid and seeks to follow the path with higher traversability values (reported by the sensors as favorable to be traversable) and avoid the cells with lower traversability values (reported by the sensors as non traversable).

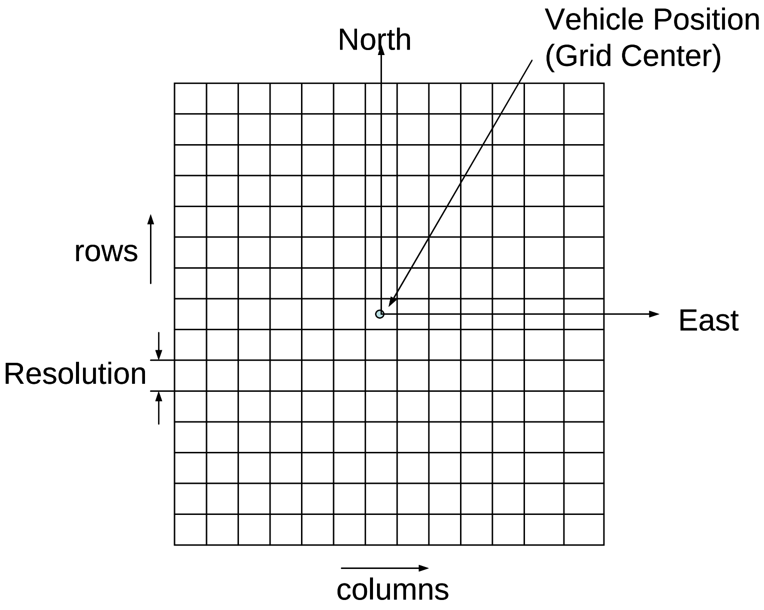


Figure 4-1. Traversability Grid representation.

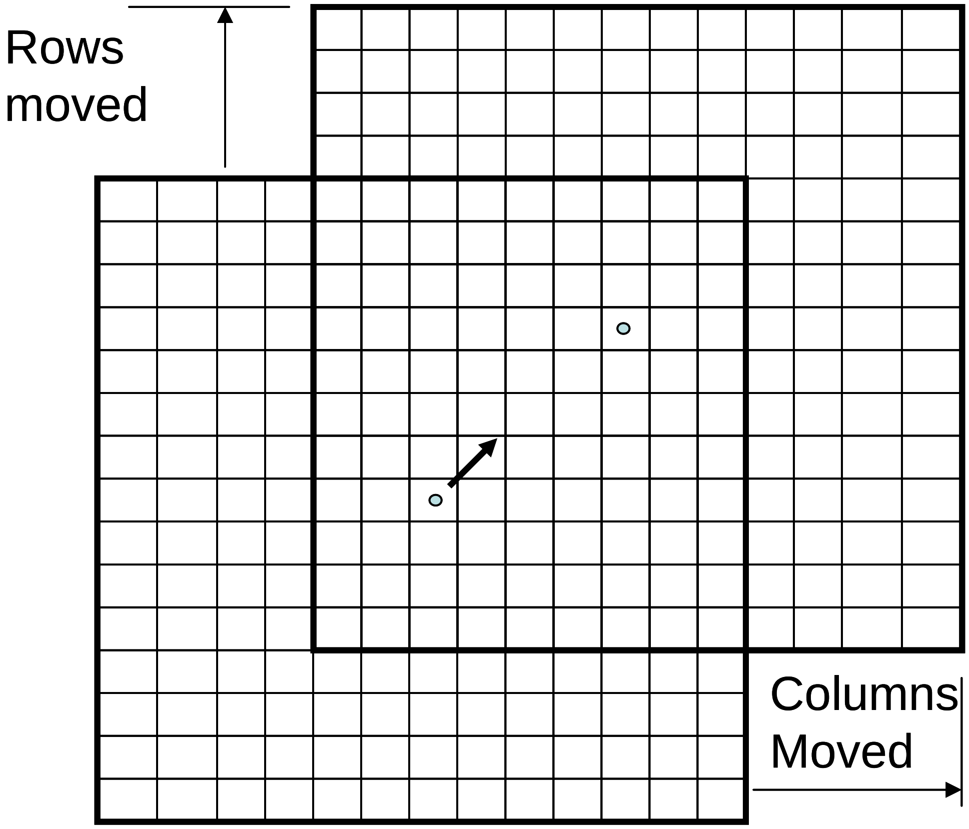


Figure 4-2. Grid movement.

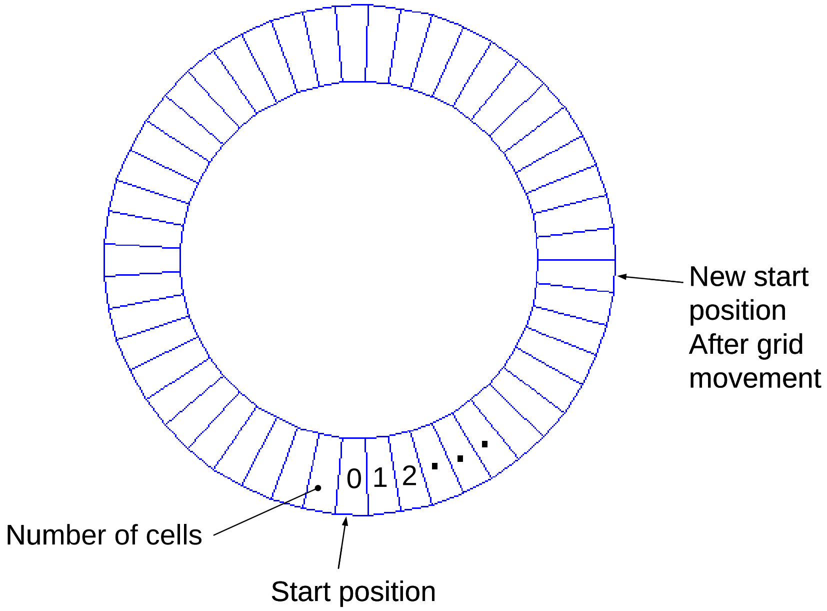


Figure 4-3. Circular buffer representation of the grid.

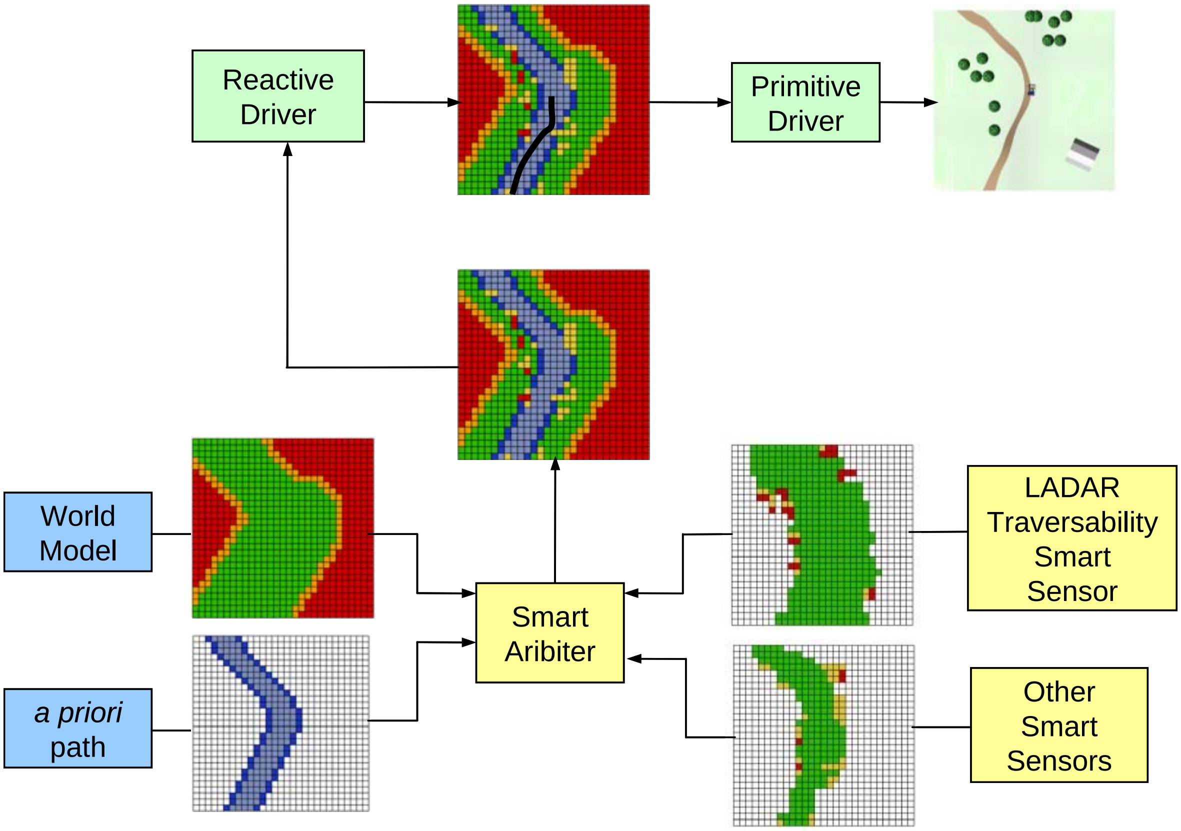


Figure 4-4. Traversability Grid propagation.

Table 4-1. Update the circular buffer to account for the grid movement.

rowsMoved and *columnsMoved* are the number of rows and columns the grid moved.

numberOfRows and *numberOfColumns* define the size of the Grid.

```
$rowBegin = rowBegin + rowMoved$
$while(rowBegin >= numberOfRows)$
{
    $rowBegin = rowBegin - numberOfRows$
}
$columnBegin = columnBegin + columnMoved$
$while(columnBegin >= numberOfColumns)$
{
    $columnBegin = columnBegin - numberOfcolumns$
}
```

Table 4-2. Accessing the grid cell in the circular buffer

```
$row = row + rowBegin$
$while\left(row >= numberOfRows\right)$
{
    $row = row - numberOfRows$
}
$column = column + columnBegin$
$while\left(column >= numberOfColumns\right)$
{
    $column = column - numberOfColumns$
}
```

The resulting (*row*, *column*) is then mapped into the 1D circular array using Equation 4.1.

CHAPTER 5
SENSOR IMPLEMENTATION

The Smart Sensor component developed to meet the research goals of this dissertation is named as the LADAR (Laser Detection and Ranging) Traversability Smart Sensor (LTSS). The LTSS component use LADAR sensors for the real-time sensing of the environment. The words LADAR and laser sensor are used interchangeably and they both mean the same. This chapter explains in detail the hardware and software implementation of the LTSS component.

Sensor Hardware

LADAR Sensor

The LADAR sensor model LMS291-S05 from Sick Inc. was selected. The LMS291-S05 is an optical sensor that scans its surrounding with infrared laser beams two dimensionally similar to laser radar. An infrared laser beam is generated by the scanner's internal diode. If the beam strikes an object, the reflection is received by the scanner and the distance is calculated based on the time of flight. The pulsed laser beam is deflected by an internal rotating mirror so that a fan shaped scan is made of the surrounding area. The laser scans an angular range of either 180° or 100°. Figure 5-1 shows the field of view for the two angular range configurations. The scanner can be configured in three angular resolutions; 1, 0.5 and 0.25°, however the resolution of 0.25° can only be achieved with a 100° range. Because of the smaller beam width, the laser is more susceptible to false echoes due to small particles. The maximum range distance measured by the system is 80 m with a measurement resolution of 10 mm. A higher range resolution of 1 mm is available with a maximum range of 8 m. The frequency of the scan depends on the angular resolution used. The scanner can operate at a maximum frequency of 72 Hz if the angular resolution is set to 1°. However for higher resolutions the operating frequency drops down to 36 Hz for a 0.5° resolution and 18 Hz for 0.25° resolution.

The range distance measured by the sensor also depends on the reflectivity of the object. For highly reflective surfaces the sensor can measure objects at greater distances then objects with low reflectivity.

Sensor Interface

The sensor operating voltage is 24 V DC +/- 15%. The 24 V power supply is fed to the sensor through the power supply system developed for the vehicle (discussed in Chapter 3). The sensor can be interfaced to the computer using serial communications. The serial interface RS232 or RS422 may be used for communication. To take advantage of the high frequency LADAR data, the data has to be transferred at a baud rate of 500 Kb. Hence the RS422 interface is selected which allows data transfer at the higher baud rates. The RS422 interface is achieved via a high speed USB to serial hub from Sealevel (part # 2403). Figure 5-2 shows the RS422 serial interface connections diagram.

Computing Resources and Operating System

The LTSS component was developed on a single computing node. The node is equipped with an AMD 2GHz processor, 512 MB ram and a 1 GB compact flash solid state hard drive. The software is developed and tested on a Linux operating system. The Linux version Fedora Core 3 was used to develop and test the software. All the software was developed using the C programming language. The GCC compiler with the built-in libraries for math functions, multithreading, socket communications and serial communications were used for the software development.

Sensor Mount

Figure 5-3 shows three LADARS mounted on the NaviGator. Two of these LADARS are mounted on a specially designed sensor cage on the top of the front end of the vehicle. The design of the mounts for these LADARS allows them to be mounted at different angular configurations. Figure 5-4, shows the design of these mounts. The LADARS are mounted facing towards the ground at different angles. These two LADARS continuously scan the ground in front of the vehicle. The third LADAR is mounted on the bumper level parallel to the ground plane.

Sensor Algorithms

So far, the concept of the representation of the environment as a Traversability Grid was discussed in Chapter 4 and the previous section discussed the actual sensor hardware used to sense the environment. Now, the mathematical approach and the software implementation of the LTSS component is explained. As mentioned in the problem statement the goal of the developed Smart Sensor component is to detect obstacles and evaluate the terrain surrounding the vehicle. To achieve this goal, the LTSS component implements a number of different algorithms.

Figure 5-5 shows a block diagram overview of the LTSS component. The input to the component is the raw range data from the lasers and the position and orientation of the vehicle and its output is the Traversability Grid which represents the state of the environment at the current time. The block diagram gives an overview of the different algorithms and flow of the outputs of each of these algorithms to form the output Traversability Grid. As shown in the Figure 5-5 the component consists of three distinct parts:

1. Obstacle Detection: The obstacle detection (OD) algorithm receives raw range data from the LADAR at the bumper level and outputs a Traversability Grid.

2. Terrain Evaluation: The terrain evaluation (TE) algorithm receives raw range data from the top two LADARS mounted on the sensor cage. Each of these two LADARS feed data into an individual terrain evaluation grid, which are then combined.

3. Sensor Fusion: The sensor fusion algorithm combines the outputs from the OD algorithm and the TE algorithm and produces a fused Traversability Grid. The fused Traversability Grid represents the results of the LTSS component and acts as an input to the Smart Arbiter like any other Smart Sensor component.

The following sections discuss in detail the different algorithms and the fusion process used to combine these algorithms.

Obstacle Detection

The LADAR sensor used for the obstacle detection (OD) algorithm is mounted at bumper level, scanning in a plane parallel to the ground at a height of 0.6 m. This height of 0.6 m may be defined as a threshold value. An obstacle of height greater than the threshold value is a positive obstacle and anything below the threshold value is free space. The LADAR is set to scan at an angular range of 180° range with a 0.5° resolution. Figure 5-1 A shows the field of view of this sensor.

The OD algorithm can be divided into two parts, the mapping of the laser range data into the Global coordinate system and then evaluating each Cell based on the mapped data.

The range data is mapped into 2D Cartesian coordinates in the Global coordinate system. As mentioned before the global coordinate system is always oriented in the North-East direction and its origin is the centerline of the vehicle at ground level below the rear axle (i.e., the projection of the GPS antenna onto the ground). After each scan of 180° the range data from the laser is converted into the Global coordinate system. This conversion is done in two steps. First the data is converted from polar coordinates to Cartesian coordinates local to the sensor as follows:

$$ {}^{S}P_0 = \begin{bmatrix} x\_sensor \\ y\_sensor \end{bmatrix} = \begin{bmatrix} \sin\alpha & 0 \\ 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} range \\ range \end{bmatrix} \qquad (5.1) $$

where

${}^{S}P_0 = \begin{bmatrix} x\_sensor \\ y\_sensor \end{bmatrix}$ is the data point in the sensor coordinate system.

The second transformation takes into account the vehicle orientation and the sensor offset distance and transforms the data point from the sensor coordinate system to the global frame.

The OD algorithm is based on the weighted sum of evidences. The traversability value of a cell is computed based on the weighted sum of the evidence of the cell being occupied or free. The evidence of a cell being an obstacle or free space is derived based on the current sensor observation and initial evidences. A sensor observation may be defined as an outcome of the sensor measurement used to evaluate the state of the system.

The sensor observation is managed internally using the two variables, '*OccupiedHits'* and '*FreeHits'* for each cell. After each laser scan the range measurements are transformed into observations. For each single coordinate generated from the range value, the cell to which this coordinate belongs in the Traversability Grid is determined, followed by all of the intervening cells between the determined cell and the sensor. Bresenham's line algorithm [Foley 1990] and [Novick 2002] is used to determine the indices of the intervening cells. The '*OccupiedHits'* buffer is incremented by one for the cell which receives the hit and the '*FreeHits'* buffer is incremented by one for all the intervening cells. For cases where the received range value is beyond the Traversability Grid map, the cell at the intersection of the line formed by the range value and the sensor origin with the bounds of the grid map is found. For all the cells on this line the '*FreeHits*' is incremented by one.

The evidence of a cell being occupied or free is computed as

$$W_{occ}\left(t\right) = W_{occ}\left(t-1\right) + OccupiedHits - k1 * FreeHits \tag{5.2}$$

$$W_{free}\left(t\right) = W_{free}\left(t-1\right) + FreeHits - k2 * OccupiedHits \tag{5.3}$$

where

k1 and *k2* are configurable parameters.

The first term in the Equations 5.2 and 5.3, is the initial weight of evidence. The initial weight of evidence defines the state of the system before the current sensor observation. These initial weights could be viewed as a Markovian model. A Markov Chain is defined as a process where the future may be stochastic but no variables prior to the current state may influence the stochastic evolution of future states. Thus, all the information of the past state of the system is represented by these initial weights.

The second term in the above equations is the sensor observations. The observation *OccupiedHits* strengthens the evidence of an obstacle and the observation *FreeHits* strengthens the evidence of free space.

The third term in both the equations above acts as a parameter to adjust the speed of response of the system. For example in the case of moving obstacles, once the obstacle is clear, the third term in Equation 5.2 helps in fast recovery of free space. It also helps in clearing spurious ground noise. Similarly, in the case where a cell is occupied by a moving obstacle the third term in Equation 5.3 helps in fast recovery of occupied space. Note however that for the case of fixed distinguishable obstacles the last terms of these equations tend to cancel each other and hence do not have a major impact on the algorithm.

After computing the weights of evidence for each cell, the weighted sum is computed as:

$$W_{sum} = W_{occ} - \rho * W_{free} \qquad (5.4)$$

where

ρ represents the ratio of evidence of a cell being occupied to a cell being free. ρ is a tunable parameter. The value of ρ for current experimental results is selected as 1/6. Finally the weighted sum value of the cell is mapped to the Traversability Value. The mapping from the W_{sum} to Traversability Value is exponential as shown in Figure 5-7.

The algorithm is very similar to the Bayes filter discussed in Chapter 2, since any algorithm that increments and decrements a state variable (in our case occupied or free), in response to sensor measurements can be interpreted as a Bayes filter. The main difference between the current algorithm and the Bayes filter is that, the Bayes filter computes the probability of one state variable, occupied or free-space. The other variable is just the negation of the computed variable. In this case both the state variables are computed and then a weighted average of these variables is performed.

The OD identifies positive obstacles and renders no opinion regarding the smoothness or traversability of areas where no positive obstacle is reported. Hence it reports Traversability values from 2 to 7. A cell with a value of 2 has a high probability of a positive obstacle while a cell with a value of 7 is free space. A traversability value higher than 7 is not assigned to a cell since it is not known if the free space is a smooth traversable path or a rough terrain or even a negative obstacle.

Terrain Evaluation

In the terrain evaluation (TE) algorithm the terrain in front of the vehicle, is mapped with the laser. A Cartesian elevation map is built from the successive laser scans and the positioning system readings corresponding to these scans. From the 3-dimensional map, the terrain is classified based on the geometry of the terrain. A set of classification features is generated by performing statistical analysis on the terrain map.

Two LADAR systems are used to map the terrain. These two LADAR’s will be identified as TerrainLADAR1 and TerrainLADAR2. The TerrainLADAR1 is mounted at an angle of 6° and the TerrainLADAR2 is mounted at an angle of 12° facing forward towards the ground. Both these LADAR’s are mounted at a height of 1.9 m above the vehicle ground level. The two LADAR’s populate two different maps of the terrain and each terrain map is classified individually. The range readings from the two different lasers are not mapped into a common terrain map, since the two lasers scan the same part of terrain at different time and hence would have a built-in GPS localization error in these two sets of readings.

The LADAR's are set to a configuration of 100° angular range and 0.25° angular resolution. Figure 5-8 shows the schematic of the field of view of the two lasers. With this configuration and for nominal conditions (flat ground surface, vehicle level), the TerrainLADAR1 scans at a distance of ~18 m ahead of the vehicle and ~43 m wide and the TerrainLADAR2 scans the ground at a distance of ~9 m ahead of the vehicle and ~21.4 m wide.

Terrain Mapping

After each complete scan of 100° the range data reported by the lasers are mapped into 3-D point clouds. The data points are stored in the corresponding cells of the Traversability Grid as a linked list of 3-D Cartesian coordinates. The conversion from the range data to the 3-D point cloud is done in the steps as explained.

The range data reported by the laser is converted into Cartesian coordinates local to the sensor. The local sensor coordinate system has its origin coincident with the origin of the laser beam and the X-Y plane coincident with the laser scanning plane. The local sensor coordinates are computed from the range information as follows:

$$\begin{aligned} x_l &= range * \sin\alpha \\ y_l &= range * \cos\alpha \\ z_l &= 0.0 \end{aligned} \tag{5.5}$$

where

α is the angle of the laser beam with respect to the y-axis of the sensor coordinate system.

Next, the local sensor coordinate is converted into the vehicle coordinate system. The vehicle coordinate system has its X-axis in the direction of the vehicle, and Z-axis vertically down.

$$\begin{aligned} x_v &= x_l * \cos\theta + x_{offset} \\ y_v &= y_l + y_{offset} \\ z_v &= x_l * \sin\theta + z_{offset} \end{aligned} \tag{5.6}$$

where

θ is the laser tilt angle with the horizontal plane and

$(x_{offset}, y_{offset}, z_{offset})$ is the position of the sensor origin in the vehicle coordinate system.

The data is then transformed into the global coordinate system attached to the vehicle as well as a fixed global coordinate system by taking into account the vehicle orientation. The transformation from the local to global coordinate system attached to the vehicle is given as follows:

$$\begin{bmatrix} x_G \\ y_G \\ z_G \\ 1 \end{bmatrix} = \begin{bmatrix} Cos\psi Cos\theta & -Sin\psi Cos\phi + Cos\psi Sin\theta Sin\phi & Sin\psi Sin\theta + Cos\psi Sin\theta Cos\phi & 0 \\ Sin\psi Cos\theta & Cos\psi Cos\phi + Sin\psi Sin\theta Sin\phi & -Cos\psi Sin\phi + Sin\psi Sin\theta Cos\phi & 0 \\ -Sin\theta & Cos\theta Sin\phi & Cos\theta Cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ z_v \\ 1 \end{bmatrix} \tag{5.7}$$

where

$\begin{bmatrix} x_G \\ y_G \\ z_G \\ 1 \end{bmatrix}$ are the global coordinates of the point,

(ψ, θ, ϕ) are the yaw, pitch and roll respectively and

(x_o, y_o, z_o) are the sensor coordinates in the global coordinate system.

The fixed global coordinate system has similar orientation as the global coordinate system attached to the vehicle, but the origin is at a fixed point whereas the origin of the global coordinate system attached to the vehicle moves with the vehicle. The fixed origin is necessary to take into account the vehicle motion to build the point cloud. The coordinates of the data point in the fixed coordinate system are computed by adding the vehicle coordinates to the global coordinates obtained above. The data point is stored in the cell corresponding to the location of the point. The coordinate system attached to the vehicle gives the location of the point in the Traversability Grid.

The maximum number of data points that can be stored in one cell is limited by a configuration variable. In case the maximum number of data points in a cell is reached, the next new data point is compared with the already existing list of data points. If the coordinates of the new data point does not match with any of the existing data points, the first data point stored in the link list is replaced by the new data point. The new data point is assumed to match with the already existing data point if it is contained in a 0.1m cube centered on the old data point. After the mapping of the data points is complete the next step is the classification of the Traversability Grid.

Terrain Classification

Each cell in the Traversability Grid is evaluated individually and classified for its traversability value. The following geometrical features are used for the classification:

1. The slope of the best fitting plane through the data points in each cell.
2. The variance of the elevation of the data points within the cell.
3. The weighted neighborhood analysis.
4. Negative obstacle algorithm.

The first three of these criteria are used for the classification of the laser data from both the LADAR's. The negative obstacle algorithm is implemented only for the data from the TerrainLADAR2.

Classification based on the slope of the best fitting plane

The slope feature helps in distinguishing discontinuous terrain or obstacle surfaces from a gradual slope. The classification of the terrain is based on the fact that the traversability of the vehicle decreases with the increase in the slope of the terrain.

The slope of the terrain is computed as the tangent of the angle between the X-Y (ground) plane and the computed plane of the terrain. The slope value is computed individually for each terrain patch corresponding to the respective cell in the Traversability Gird. Based on the data points in the cell, the terrain patch is approximated as a planar surface using the least squares error approximation.

To approximate the plane of the terrain patch a minimum of three data points must be present in the cell. The equation for the best fitting plane, derived using the least squares solution technique, is given as:

$$S_{optimum} = \left(G^T G\right)^{-1} G^T b \tag{5.8}$$

where:

$S_{optimum}$ is the vector $(S_{x,} S_{y,} S_z)$ perpendicular to the best fitting plane

G is an n × 3 matrix given by:

$$G = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ - & - & - \\ x_n & y_n & y_n \end{bmatrix}$$

b is a vector of length 'n' given by:

$$b = \begin{bmatrix} -D_{01} \\ -D_{02} \\ - \\ -D_{0n} \end{bmatrix} \quad .$$

Assuming D_{0i} equal to 1, Equation 5.8, is used to find $S_{optimum}$ for the data points within each cell. Chapter 5 of [Solanki 2003] provides a thorough proof of the Equation 5.8 for finding the perpendicular to a best fitting plane given a set of 3-D data points. Once the vector perpendicular to the best fitting plane is known, the slope of the computed plane with respect to the X-Y (ground) plane is computed as follows:

$$Slope = \frac{\sqrt{\left(S_x{}^2 + S_y{}^2\right)}}{S_z} \tag{5.9}$$

The above computed slope value represents the tangent of the angle between the best fitting plane and the ground plane. This value is used to assign the traversability value. The assignment of the traversability value is heuristic and based on comparing the classification results with the actual terrain conditions. Table A-1 lists the mapping of the slope value to the traversability value used in the experiments for this research. Instead of showing the tangent of the angle, the Table A-1, shows the value of the angle between the best fitting plane and the ground plane.

Classification based on the variance

The variance is a measure of the dispersion of data. The dispersion of data is defined as the extent to which the data is scattered about the zone of central tendency. The variance is defined as the sum of the squares of the deviations from the mean value divided by the number of observations. For the terrain classification problem, the variance of the data in the *Z* direction is an important parameter. The variance of the data points in the Z direction gives an indication of the traversability of the terrain. A higher value of the variance indicates a rougher, uneven and hence less traversable terrain while decrease in the variance indicates a smoother, traversable terrain condition. From the data points within each cell, the variance of the terrain patch corresponding to the respective cell is measured as follows:

$$\mathrm{var}\,iance = \frac{\Sigma\left(Z_i - \mu\right)^2}{n} \qquad (5.10)$$

where μ is the mean height of the cell given as:

$$\mu = \frac{\Sigma Z_i}{n} \qquad (5.11)$$

Z_i is the elevation of the data point, and

n is the number of data points.

The variance value computed using the Equation 5.11, is mapped to a traversability value which is assigned to the cell. The mapping from the variance value to the traversability value is heuristic and based on classification results obtained from driving through different terrain conditions. The mapping variables are stored as configuration parameters in the config file. The mapping values used in the experimental results of this dissertation are shown in Table A-2.

Weighted Neighborhood Analysis

As discussed in Chapter 2, a few research papers [Ye 2004] use the mean height as a criterion for evaluating the terrain. However, using absolute height as a parameter to evaluate terrain often results in misclassification of the terrain especially in case of uphill and downhill slopes. Instead of evaluating terrain based on the absolute height of each cell, a measure of terrain evaluation presented here is to compare the height of neighboring cells. The comparison of the neighboring cells gives an indication of the discontinuity in the terrain between the two cells.

One of the difficult problems in neighborhood analysis is determining the extent of neighboring cells to use for comparison. Figure 5-9 shows the application of neighborhood analysis in the present context. Assuming that the current position of the vehicle is on a traversable region, consider the mean height of the center cell of the grid as ideal (the vehicle is always in the center of the grid). Now, move out from the center cell and compare the mean height of each of the cells adjacent to the center cell with the height of the center cell. As the cells examined expand from the center compare the height of each cell to be evaluated with the neighboring cells that are between the vehicle (i.e., the center cell) and the cell being evaluated. The idea here is that since these neighboring cells fall in between the vehicle position and the cell being evaluated, for the vehicle to travel through the cell, it has to pass through these neighboring cells. Hence if the terrain is discontinuous between the cell under consideration and the neighboring cells, then the cell is less likely to be traversable. For any cell in the grid there are three neighboring cells as shown in Figure 5-9, which fall in between the cell and the center of the grid. These three neighboring cells are the cell in the adjacent row, the cell in the adjacent column and the diagonally adjacent cell. Since there are more than one cell which is adjacent to the cell being evaluated an algorithm to decide on the importance of each neighboring cell is designed. The algorithm takes into consideration, that the importance of each of the individual neighboring cells is dependent on the position of the cell under consideration in the grid. For example, for a cell which is in the diagonal direction of the grid the diagonally neighboring cell will have more weight age then the neighboring row and column cell. Similarly the cells which are towards the center row of the grid will have more weight age on the neighboring column cell and the cells towards the center column of the grid will have more weight age on the neighboring row cell. Each neighboring cell is assigned a weight, which is computed based on the position of the cell in the grid.

For any cell in the grid, compute the unit vector in the direction from the cell to the center of the grid. The neighboring cells are the adjacent cells that are closer to the center. These are assigned weights depending on the position of the vector. Consider a unit vector, $\overline{v} = a\hat{i} + b\hat{j}$, which represents the direction from a cell to be evaluated to the center of the grid (vehicle position). The *i* component of the vector is in the direction of the neighboring row cell and the *j* component is in the direction of the neighboring column cell discussed above. The neighboring row, column and diagonal cell are assigned weights *c1*, *c2*, and *c3* as follows:

$$\begin{aligned} c1 &= \frac{a}{\left(a+b+\frac{1}{\sqrt{2}}(a+b)\right)} \\ c2 &= \frac{b}{\left(a+b+\frac{1}{\sqrt{2}}(a+b)\right)} \\ c3 &= \frac{\frac{1}{\sqrt{2}}(a+b)}{\left(a+b+\frac{1}{\sqrt{2}}(a+b)\right)} \end{aligned} \tag{5.12}$$

From the Equations 5.12, it can be seen that the weights are chosen based on the components of the unit vector in each of the directions. These weights are normalized so that the sum of the weights is always 1.

A weighted neighborhood analysis is then done for the cell being evaluated. Thus, depending on the height difference between the cell being evaluated and the neighboring cells, a traversability value is assigned to the cell. For example, consider the cell shown in Figure 5-9. The neighborhood cell analysis value is calculated as follows:

$$Value(i,j) = c1*\left[h(i,j) - h(i-1,j)\right] + c2*\left[h(i,j) - h(i-1,j-1)\right] + c3*\left[h(i,j) - h(i,j-1)\right]$$

(5.13)

where

$h(i,j)$ is the height of the *cell* in *row i* and *column j*.

The above computation is shown for a cell in the first quadrant of the grid. Similarly for the cells lying in the different quadrants of the grid, the corresponding neighboring cells are selected using the scheme discussed above. The cells in the center row and the center column are treated as special cases and the neighborhood analysis of these cells is done by comparing them with only one cell, which is the neighboring cell directly in between the cell being evaluated and the center of the grid.

The neighborhood analysis value obtained for each cell from the above calculations is mapped to a traversability value. Again in this case the mapping is heuristic and is saved as configuration parameters in the config file. Table A-3 shows the mapping of the Weighted Neighborhood analysis value to the traversability value used in the current experimental results.

Negative Obstacle Detection

In a rough outdoor environment, negative obstacles such as big holes or cliffs on the side of the road are very common. With the laser range sensor, the only information one can obtain is the distance of the object (terrain surface or obstacle) hit by the laser beam. With this information, the terrain map was built and algorithms were developed to classify terrain. However in case of voids, empty space, the only important information that can be used from the laser data is that the region in between the laser and the laser range reading is a free space. The negative obstacle algorithm makes use of this information to give an estimate if the free space is negative obstacle.

The algorithm compares each range reading obtained from the laser to the expected range reading. The expected range reading is computed based on the geometry of the laser beam. The unit vector in the direction of the laser beam is known since the laser is fixed with respect to the vehicle and the vehicle orientation is known. Expecting good terrain condition, it is assumed that the region in front of the vehicle is a level ground in the plane of the vehicle. From the geometry of the laser beam (i.e., the line vector in the direction of the laser beam) and the assumption of a level ground (X-Y plane) the expected range reading is computed by solving the problem of the intersection of a plane and a line.

Consider a unit vector in the direction of the laser beam. The unit vector is expressed in the Laser coordinate system using Equation 5.5. The unit vector is then transformed to the vehicle orientation and subsequently to the global orientation using Equations 5.6 and 5.7. The sensor offset terms in Equation 5.6 are not considered, since it is only the direction of the vector which needs to be expressed in the global orientation frame.

The expected range distance in the direction of the unit vector is computed using the information that the z-component of the expected range vector expressed in the global orientation frame with the origin attached to the sensor is equal to the height of the laser above the ground plane. The expected range is given as:

$$d = h / \left(e_r \bullet \hat{k} \right) \tag{5.14}$$

where

d is the range distance from the sensor origin to the point of intersection,

h is the height of the sensor above the vehicle ground plane and

e_r is the unit vector representing the direction of the laser beam in the global coordinate system.

The expected range reading is subtracted from the actual reading obtained from the laser and the difference is used as a factor to assign traversability value. If the difference is less than a threshold value, the algorithm concludes that ground exists (positive obstacle or smooth flat terrain) and does not report anything. However, if the difference is above the threshold value, the possibility of a negative obstacle is assumed. The cell in the Traversability Grid which would otherwise register the expected reading (in case of flat ground plane) is assigned a traversability value. This cell would lie on an imaginary plane formed as an extension of the vehicle ground plane similar to the one shown in Figure 5-10. The cell can easily be found since the range vector in the Global coordinate system is already known from the Equation 5.14 and it just needs to be transformed from the sensor origin to the vehicle origin (center of the Traversability Grid).

The severity of assigning this cell as a negative obstacle; increases with the increase in the difference in the actual and expected range distance readings. The algorithm assigns a traversability value between 2 and 7, since it is seeks only negative obstacles and does not render any opinion on smoothness or good terrain conditions.

The algorithm is very sensitive to the tilt angle of the laser with respect to the ground plane. The algorithm cannot distinguish between the severities of negative obstacle if the change in the slope of the ground plane is greater than the tilt angle of the laser. The TerrainLADAR2 sensor is tilted at an angle of 12° towards the ground and the TerrainLADAR1 is tilted at an angle of 6° towards the ground. TerrainLADAR2 sensor would be able to distinguish between traversable path and negative obstacle as long as the change in the slope of the traversable path is less then 12° and if this change is greater then 12°, there is a high probability of a false positive being registered. For the TerrainLADAR1 this allowable change in slope is limited to only 6°. Hence, the above algorithm is implemented only for the data from the TerrainLADAR2 sensor.

The probability of registering a false positive also depends on the vehicle orientation. For example, in cases where the vehicle is on a horizontal ground plane and there is a down hill slope in front of the laser, with magnitude greater than the laser tilt angle, the algorithm would indicate a negative obstacle irrespective of whether the slope is a gradual, smooth downhill slope or it is a sudden discontinuity in the terrain. However, if the vehicle orientation is pitched up i.e., the vehicle is on an uphill slope, the output from the algorithm would depend on the vehicle orientation angles, since the direction of the laser beam depends on the vehicle orientation angles.

The negative obstacle detection algorithm assigns traversability values only to the cells, which do not register any data points from the terrain mapping algorithm. If there are data points present in the cell, the cell is not considered to be evaluated for negative obstacle.

Terrain Evaluation Output

The previous sections discussed the four algorithms used to evaluate the terrain characteristics. These algorithms are implemented separately on each of the two sensor data. Figure 5-5, showed the overall block diagram of the LTSS component. From the figure it can be seen that, the terrain is evaluated by processing the data from each of the Terrain LADAR's separately. The blocks terrain evaluation 1 and terrain evaluation 2, represent the terrain evaluation process for the two LADAR's. A more elaborate picture of the terrain evaluation 1 block is shown in the Figure 5-11. As discussed in the previous sections and as shown in the figure, the laser data is first mapped into a terrain model. This terrain map is then input to each of the terrain evaluation algorithms. The Traversability Grid output from the terrain evaluation 1 block is the fusion of the traversability values obtained from each of the algorithms.

The slope and the variance algorithms evaluate the cell based on the data across each single cell. Each of these algorithms would work better than the other depending on the terrain condition being evaluated. The average of the slope and variance based traversability values gives an evaluation of the cell based on the data within the cell. The nearest neighborhood analysis evaluates the cell based on the discontinuity in the data between cells. The output of the terrain evaluation 1 block is the minimum of the value above two values.

The traversability values from the terrain evaluation algorithm implemented on the TerrainLADAR2 is obtained using a similar equation as 5.15. Except in the case of TerrainLADAR2, if there is no data present in a particular cell and that cell has been evaluated by the negative obstacle detection algorithm, the cell would be assigned a traversability value from the negative obstacle detection algorithm. All the other cells which do not contain any data and which have not been evaluated by the negative obstacle detection algorithm are considered to be not in the field of view of the sensor and hence traversability value of 14 (unknown) is assigned to the cell.

A simple averaging algorithm combines the outputs of the terrain evaluation 1 and terrain evaluation 2. The computed average is the traversability value of the cell based on the terrain evaluation:

$$TTV = \frac{TTV_1 + TTV_2}{2} \qquad 5.16$$

where,

TTV is the traversability value of the cell based on terrain evaluation,

TTV_1 is the traversability value of the cell assigned by the terrain evaluation 1,

TTV_2 is the traversability value of the cell assigned by the terrain evaluation 2.

Advantages and Limitations of the OD and TE Sensor Algorithms

The experiments conducted with the above sensor algorithms reveal the following advantages and limitations

1. The obstacle detection (OD) algorithm can be used only for positive obstacles; it gives no opinion on the smoothness of the terrain and negative obstacles.

2. In off-road conditions the OD generates a lot of ground noise. Most of the noise is due to misclassification of an approaching uphill slope or due to going down a hill transitioning into flat ground.

3. However the OD algorithm is not as complex as the terrain evaluation (TE) and it does not have to create a 3-D point cloud. It is very reliable in identifying positive obstacles. The grid is updated about 35 times a second in the region bounded by the field of view of the sensor. In this region moving obstacles which have passed are cleared in the grid and if a moving obstacle shows up in the grid, the grid will be updated.

4. Since the OD algorithm does not depend on mapping the true coordinate of the point; but just checks to see if the point belongs to a cell, the error in mapping the obstacle is very small compared to the TE algorithm

5. The main concern with the TE algorithms is modeling the ground plane. Since the data is collected in successive scans, the ground plane of the vehicle changes. Thus each time the points are registered with a different reference plane. In cases where the vehicle is on flat smooth terrain this is not a problem. However, in cases of uneven terrain, it is very difficult to relate the data in a common ground plane. Although the points are registered in a fixed global frame there is some error associated with the registration process and experiments have shown the magnitude of the error depends on the condition of the terrain.

6. Since the look ahead distance of the Terrain LADAR's is limited by the tilt angle of the laser, in the present case TE algorithm is effective for a range of only up to 18 m. It does not provide any information of obstacles further than this distance.

7. The TE algorithm maps moving obstacle as part of the terrain and hence does not clear them after they have passed from the grid.

8. In spite of the above disadvantages of the TE, the algorithm actually maps the surrounding into a 3D point cloud and characterizes the terrain based on slope, variance and discontinuities, and hence the classification is based on more detailed information of the surrounding as compared to the OD.

The conclusion that can be drawn from the discussion and actual implementation of the OD and TE sensors is that these sensors provide very good classification results in a limited range of environmental conditions. However, much uncertainty is associated with these two sensor implementations, when using them in the real world heterogeneous environment. The following section presents an uncertainty management tool, which is used to fuse the outputs from these two algorithms.

Fusion of the Sensor Components

In the previous sections two different sensor algorithms, the OD and the TE were developed and the advantages and disadvantages of each were discussed. To take advantage of each sensor algorithm and at the same time overcome some of its limitations, the outputs from the above sensors are fused together using a simple rule-based forward reasoning scheme. The uncertainties associated with the two sensors are combined using certainty factors [4]. The certainty factor (CF) formalism presents an approach to combine evidences supporting or contradicting a hypothesis. As opposed to the Bayesian analysis where only those evidences supporting a hypothesis can be combined, the certainty factor formalism provides a mechanism to combine contradictory evidences. The certainty factor value for a particular hypothesis is between -1.0 and 1.0. The CF value of 1.0 represents complete confidence on the hypothesis while a CF value of -1.0 represents a complete confidence against the hypothesis. In the present case, a CF value of 1.0 indicates complete confidence in the presence of an obstacle or highly non traversable region while a CF value of -1.0 represent a highly traversable and smooth region. The value of 0.0 represents a cell which does not show any confidence either towards the presence of an obstacle or towards a desirable traversable path. The rule based reasoning scheme assigns different CF values to each of the sensors based on the observed readings. Information such as the mean height of the cell is used in determining the confidence level whether the sensor should be able to see the obstacle.

The traversability values obtained from each of the sensors is converted into the evidence of the presence of an obstacle or presence of a traversable path. The confidence on the evidences presented by each of these sensors is represented as,

$CF_{evidenceOD}$ is the evidence of the presence of an obstacle detected by the OD algorithm. As discussed in the previous sections, since the OD sensor does not give any input on the traversable path, the value of this variable is between 0.0 (representing traversability value of 7) and 1.0 (representing traversability value of 2) depending on the presence of Obstacle.

$CF_{evidenceTE}$ is the evidence of the traversability or non-traversability of the cell detected by the TE sensor. The value of this variable is between -1.0 (traversability value of 12) and 1.0 (traversability value of 2) depending on the traversability of the cell.

The pseudo code for the implementation of the fusion of these two sensor algorithms is presented below.

CF_{OD} and CF_{TE} are the certainties associated with the respective sensors depending on which rule executes.

CF_{LTSS} is the combined CF value.

The following scheme is applied:

Case 1: The OD indicates the Cell is OCCUPIED

```
IF      TE = UNKNOWN
THEN
        $CF_{OD} = 1.0 \times CF_{evidenceOD}$
        $CF_{LTSS} = CF_{OD}$

IF      TE = NON-TRAVERSABLE
THEN
        $CF_{OD} = 1.0 \times CF_{evidenceOD}$
        $CF_{TE} = 1.0 \times CF_{evidenceTE}$
        $CF_{LTSS} = CF_{OD} + CF_{TE} \times (1 - CF_{OD})$


IF      TE = TRAVERSABLE
THEN
```

$$CF_{OD} = 0.9 \times CF_{evidenceOD}$$

$$CF_{TE} = 0.9 \times CF_{evidenceTE}$$

$$CF_{LTSS} = \frac{CF_{OD} + CF_{TE}}{1 - \min\left(\left|CF_{OD}\right|, \left|CF_{TE}\right|\right)}$$

Case 2: The OD indicates the Cell is FREE

```
IF      TE = UNKNOWN
THEN
        $CF_{LTSS} = CF_{OD}$

IF      TE = NON-TRAVERSABLE
THEN
        IF      (Mean Height <= 0.6m)
        THEN    $CF_{TE} = 1.0 \times CF_{evidenceTE}$
        IF      (Mean Height > 0.6m && Mean Height < 0.8m)
        THEN    $CF_{TE} = 0.8 \times CF_{evidenceTE}$
        IF      (Mean Height >= 0.8m)
        THEN    $CF_{TE} = 0.2 \times CF_{evidenceTE}$

        $CF_{LTSS} = CF_{TE}$

IF      TE = TRAVERSABLE
THEN
        $CF_{TE} = 1.0 \times CF_{evidenceTE}$
        $CF_{LTSS} = CF_{TE}$
```

Case 3: The OD indicates the Cell is UNKNOWN

$$CF_{TE} = 1.0 \times CF_{evidenceTE}$$

$$CF_{LTSS} = CF_{TE}$$

The CF values are mapped back into the traversability value. As discussed earlier, -1.0 corresponds to a value of 12, with 7 corresponding to a value of 0 and 1.0 corresponding to a value of 2. Should one of the sensors fail to report, all the values in the grid for that sensor are marked as unknown, and the above scheme would give the output from the other sensor as the fused output. Hence, the fusion process is modular in the sense that each one of the sensors output would still be valid as a final output should the other sensor fail.

Implementation of the LTSS as a JAUS component

The LTSS is implemented as a JAUS component. This section starts with a brief overview of the JAUS architecture followed by the JAUS implementation on the autonomous test platform, NaviGator. Finally a detailed explanation of the implementation of the LTSS as an experimental JAUS component is given.

Joint Architecture for Unmanned Systems

The Joint Architecture for Unmanned Systems (JAUS) [JAUS] is an architecture defined for use in the research, development and acquisition of unmanned systems. The two main purposes for the development of JAUS are to support interoperability amongst heterogeneous unmanned systems originating from different developers and to support the reuse/insertion of technology. To ensure that the architecture is applicable to the development of entire domain of unmanned systems the following constraints are imposed on JAUS; platform independence, mission isolation, computer hardware independence and technology independence. JAUS is a component based, message passing architecture that specifies data formats and methods of communication among computing nodes. The JAUS system architecture is defined in a hierarchical structure. The system topology is shown in the Figure 5-12. The different levels of the architecture are defined in the following terms:

System: A system is comprised of all the unmanned systems and human interfaces meant for a common application.

Subsystem: A subsystem is a single or more than one unmanned system which can be defined as a single localized entity within a system. The autonomous platform, Navigator which has been developed at CIMAR may be defined as a single JAUS subsystem

Node: A JAUS node defines a distinct processing capability within the subsystem. Each node runs its own node manager component to manage the flow and control of JAUS messages.

Component: A component provides a unique functional capability for the unmanned system. A JAUS component resides wholly within a JAUS node. More then one components may reside on a single node.

Instances: Instances provide a way to duplicate JAUS components. All components are uniquely addressed using the subsystem, node, component and instance identifiers.

JAUS defines a set of reusable components and the messages supporting these components. However, JAUS does not impose any regulations on the configuration of the system. JAUS also allows the development of experimental components for performing tasks which otherwise cannot be performed by the already defined JAUS components. The only absolutely necessary requirement that has to be satisfied for the implementation of JAUS is that all JAUS components can communicate between each other only through JAUS messages.

JAUS System Architecture on the NaviGator

In the hierarchical structure of the JAUS system, the NaviGator is defined as a fully independent JAUS subsystem. The NaviGATOR system architecture is formulated using the existing JAUS defined components wherever possible. Experimental JAUS components are developed for the tasks which did not have a JAUS component. A JAUS compliant messaging system is used to define all the communication between components. The sensor component developed in this research is an experimental JAUS component.

Each of the rectangular blocks shown in Figure 5-13, is a JAUS component. From the autonomous functionality view point, at the highest level, the Navigator system architecture is categorized into four fundamental elements. These elements are:

1. Planning element: The components that act as a repository for a priori data. These components perform off-line planning based on the a priori data.
2. Control element: The components that perform closed loop control in order to keep the vehicle on a specified path.
3. Perception element: The components that perform the sensing tasks required to locate obstacles and to evaluate the smoothness of terrain.
4. Intelligence element: The components that determine the best path segment based on the sensed information.

As stated in Chapter 4, the Traversability Grid is the common data structure used to represent the environment in all the above components. The components in the perception element represent the world as a Traversability Grid based on real-time perception, the components in the planning element represent the world around the robot as a Traversability Grid based on a priori information, the components in the intelligence element utilizes these Traversability Grids to plan the best possible path and the control element executes the planned path. The complete loop of perception, planning and control is repeated continuously at about 40 Hz. The following section explains in detail the implementation of the LTSS as a JAUS component which comprises of all the sensor algorithms discussed above.

Implementation of the LTSS as a JAUS component

The LTSS JAUS component is developed using C programming language in the Linux Operating system environment. The LTSS and all the other JAUS components on the NaviGator are implemented as finite state machines. At any point of time, each component can assume one of the seven states enumerated as; Startup, Initialize, Standby, Ready, Emergency, Failure and Shutdown. Of these the Emergency state is not used in the LTSS component; the Failure state is used to report any type of failure, such as failure to allocate dynamic memory or failure to create service connections and the Shutdown state is called during shutdown of the component to end all the processes in a proper sequence. The important operations in the other states are in the following sequence:

Startup state: The component is checked into the system and obtains instance, node and subsystem identification numbers from the node manager. All the required data structures declared as global variables are initialized using dynamic memory allocation. These include three data structures of type Circular Buffer to store the grid information for each of the LADAR's. The basic implementation of the circular buffer was discussed in Chapter 3. Since the obstacle detection algorithm and the terrain evaluation algorithm store variables of different data types, the circular buffer implementation for the two; is different in regards to the number and type of variables but the basic functioning remains the same. The position and orientation information is obtained from the JAUS component, GPOS and the vehicle state information is obtained from the JAUS component, VSS. This information is obtained using JAUS service connections. The JAUS service connections provide a mechanism to continuously obtain information at a fixed update rate from another component without the necessity to query the component each time for the information. Before entering into the Initialize state, the LTSS component creates the service connections to the GPOS and the VSS in the Startup state. Finally all the config variables in the config file are loaded in the program.

Initialize State: This state makes sure that the GPOS and VSS service connections are active. In case these connections are not active the component remains in the initialize state. Even when in Ready state, if the GPOS or VSS service connections are down the component defaults to initialize state.

Ready state: This is the most important part of the component and all the processing is done in this state. Once the component is set to run with all the global variables initialized and the service connections active, it remains in the ready state. It loops through the ready state once every time it produces an output Traversability Grid. The following sequence of steps is performed each time:

1. Conversion of the vehicle position from the LLA (Latitude/Longitude/Altitude) data format to the vehicle position in a fixed Global coordinate system using the UTM (Universal Transverse Mercator) conversion. This information is then transformed into the number of rows and columns moved by the vehicle since the previous update.

2. Update each of the Circular buffers to account for the number of rows and columns moved by the vehicle. The circular buffer update method to account for the movement of the vehicle was discussed in Chapter 3.

3. The next step is to acquire the range data from each of the lasers. The data from each laser is acquired in a different thread. The coordination between the laser data acquisition thread and the main component thread is maintained using a mutex lock. Since the OD laser runs at a frequency of 36 Hz and the two TE laser's run at a frequency of 18 Hz., the OD laser loads a new set of range readings for every iteration of the Ready state, while each of the Terrain evaluation lasers alternatively load a new set of readings for every iteration.

4. After the acquisition of the range data, the aforementioned algorithms pertaining to the mapping of data and assignment of the traversability values are implemented. The algorithms are executed in the following sequence; the Traversability Grid for the OD is updated with the new data, the two Traversability Grids for terrain evaluation 1 and terrain evaluation 2 are populated with the most recent data from the two terrain lasers respectively. Next, the OD algorithm evaluates and assigns each cell a traversability value, similarly the two terrain evaluation Traversability Grids are evaluated and traversability values of these two terrain evaluation grids are then combined into a traversability value. The final step is the fusion of the OD and the TE grid.

5. The output Traversability Grid is passed on to the Smart Arbiter. The algorithm repeats the steps in the Ready state as long as the service connections for the GPOS and VSS are active.

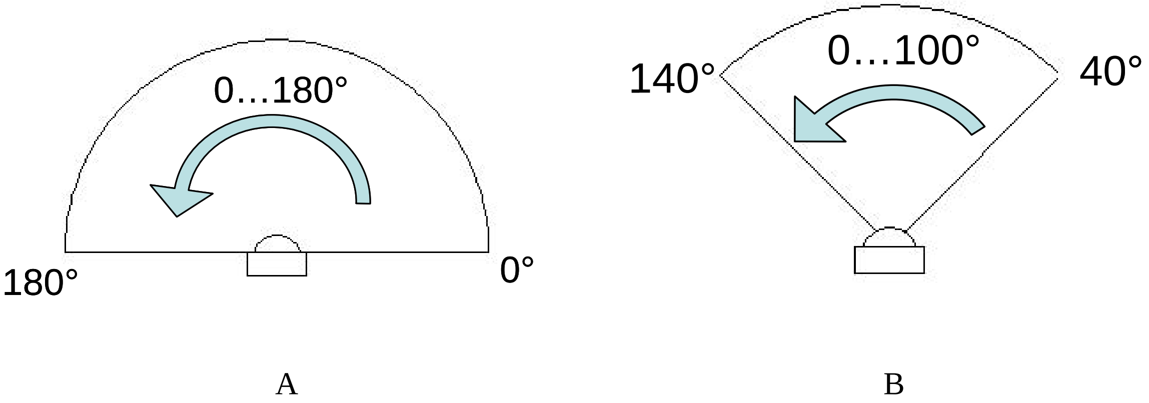
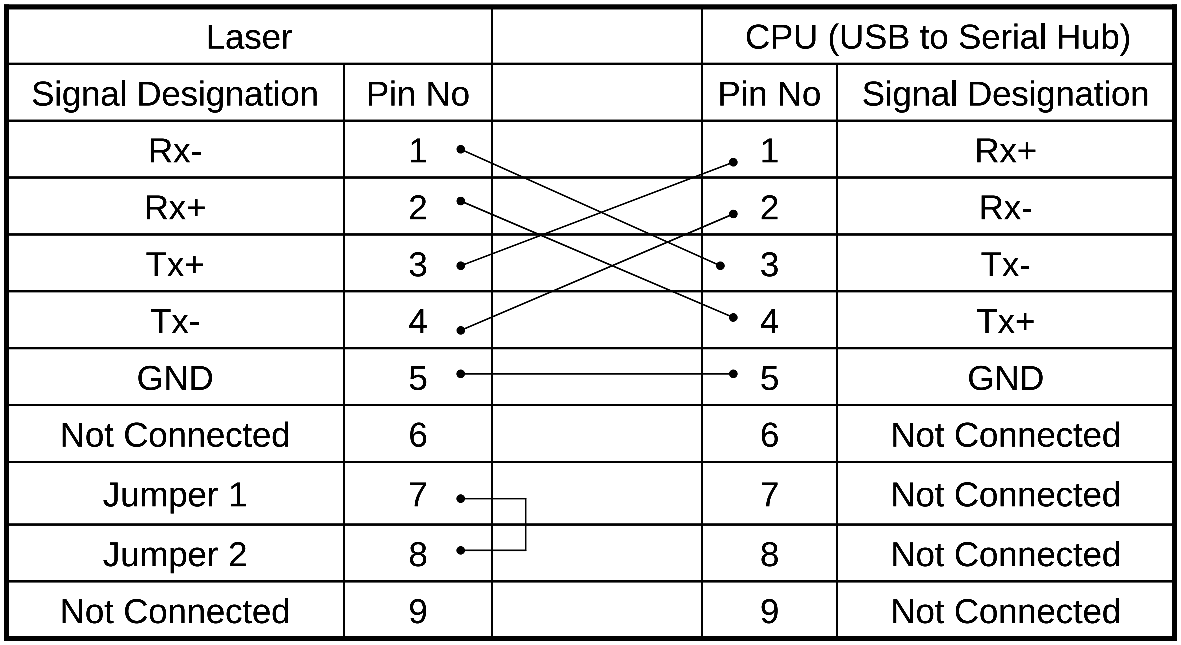


Figure 5-1. Measurement range (Top view scan from right to left). A) Angular Range 0° -180° B) Angular Range 0°-100°.



Laser			CPU (USB to Serial Hub)	
Signal Designation	Pin No		Pin No	Signal Designation
Rx-	1		1	Rx+
Rx+	2		2	Rx-
Tx+	3		3	Tx-
Tx-	4		4	Tx+
GND	5		5	GND
Not Connected	6		6	Not Connected
Jumper 1	7		7	Not Connected
Jumper 2	8		8	Not Connected
Not Connected	9		9	Not Connected

Figure 5-2. Laser sensor RS422 interface

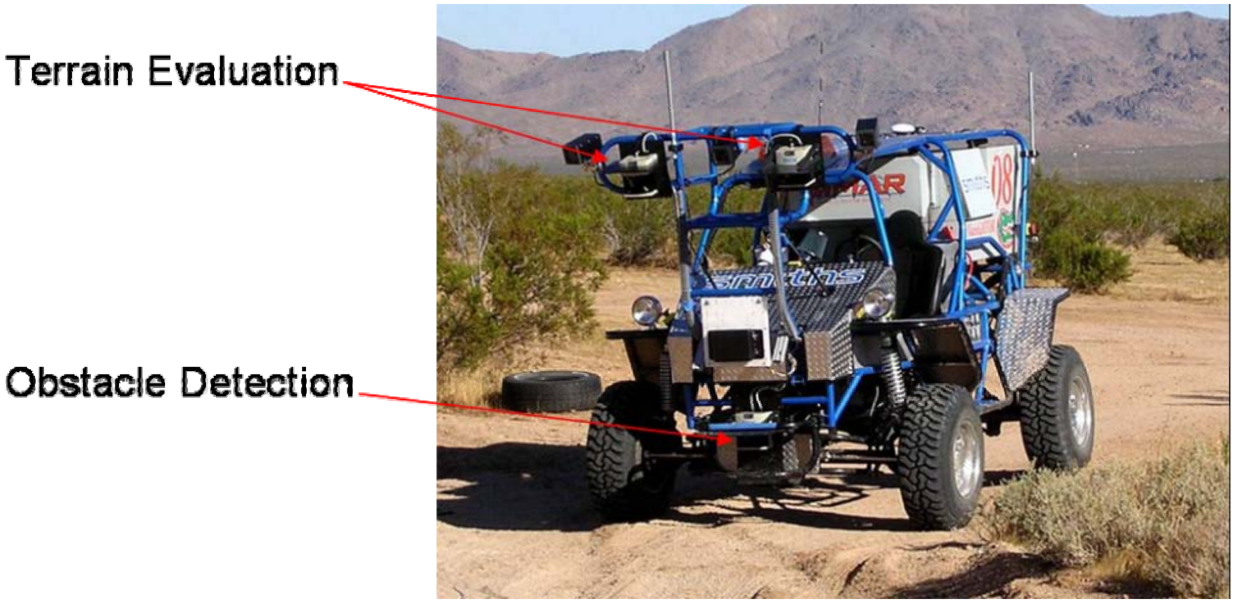


Figure 5-3. Sensor mounts on the NaviGator.

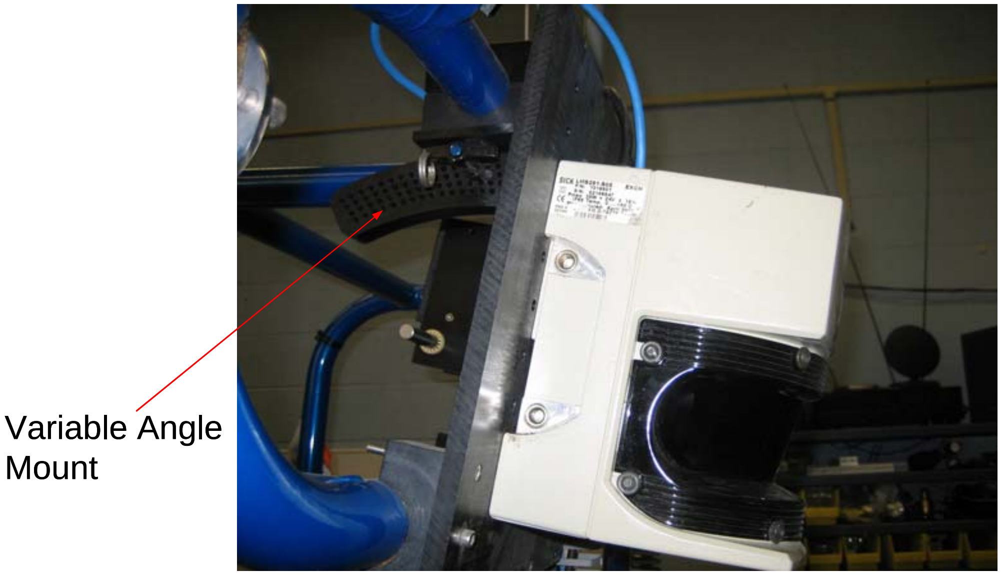


Figure 5-4. Sensor mount design.

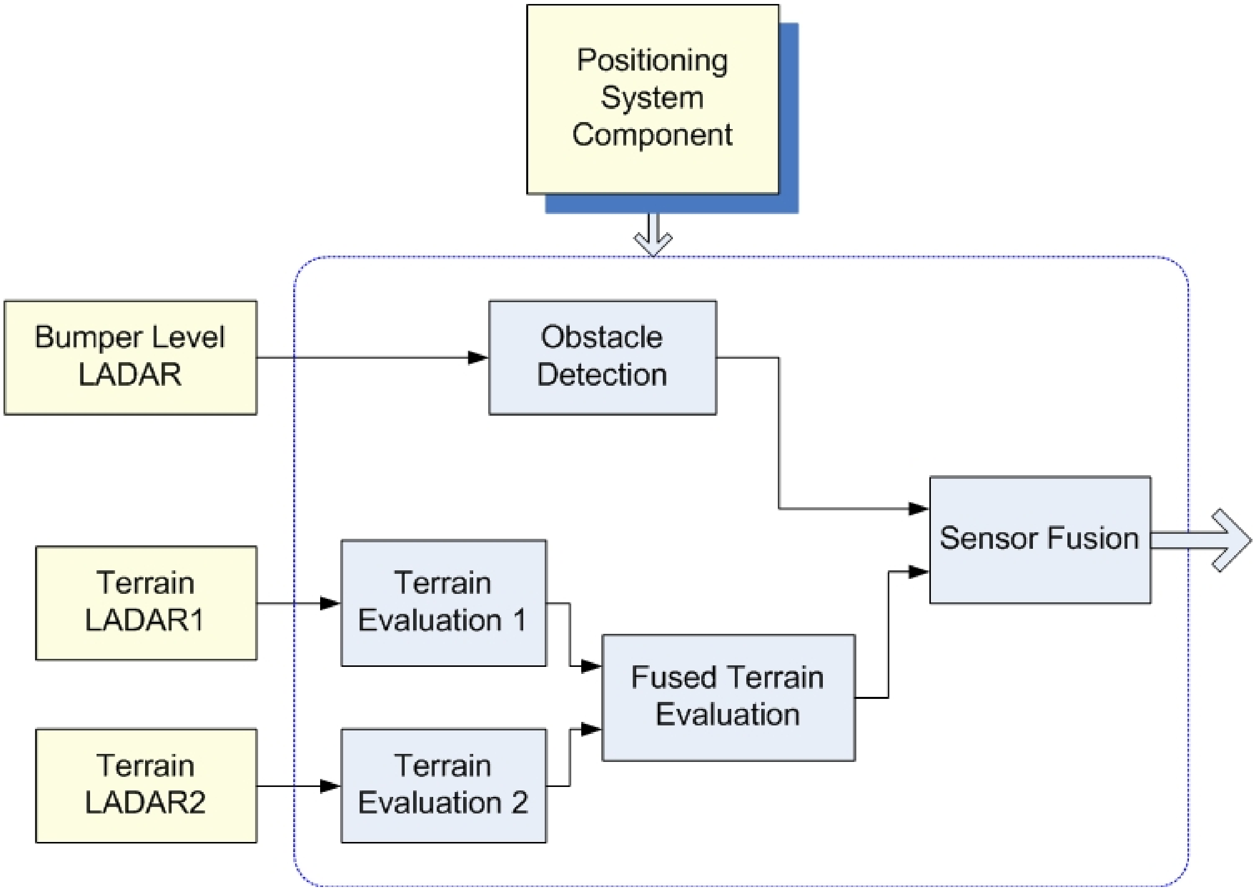


Figure 5-5. Block diagram of the LTSS component

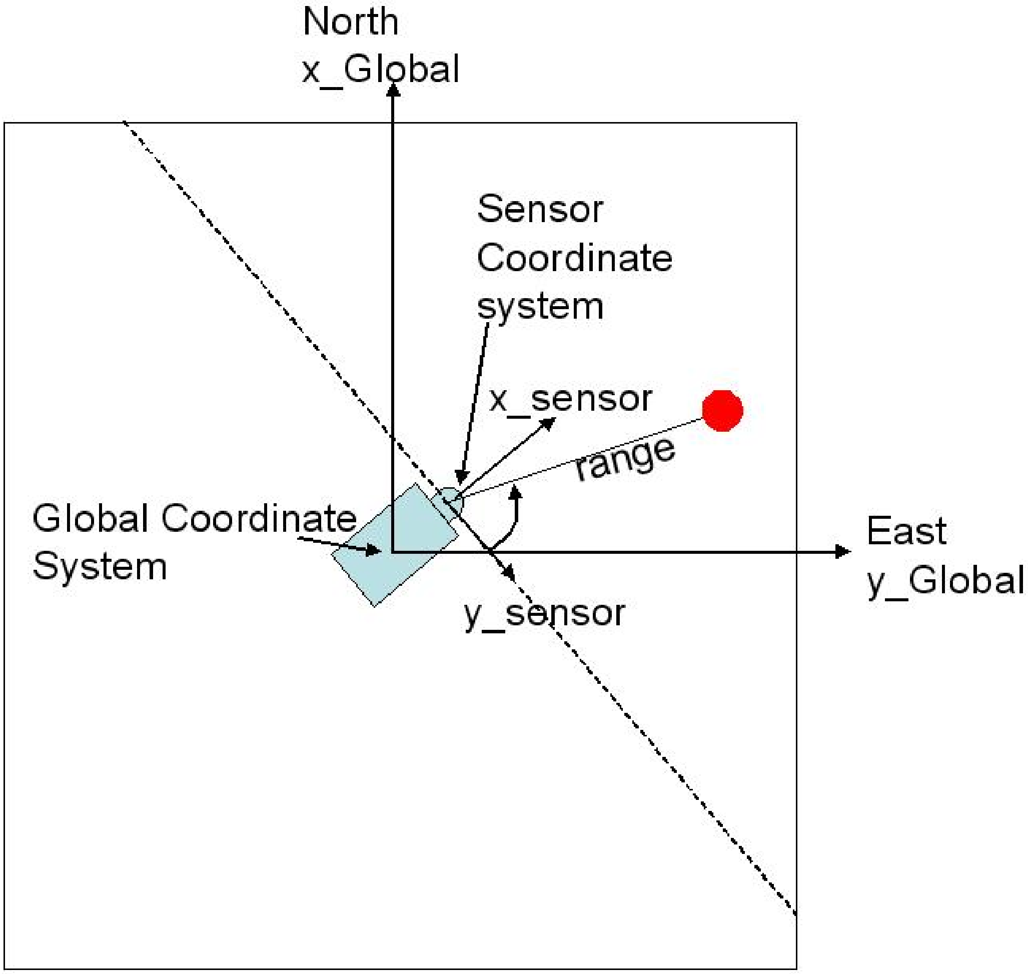


Figure 5-6. Obstacle detection LADAR.

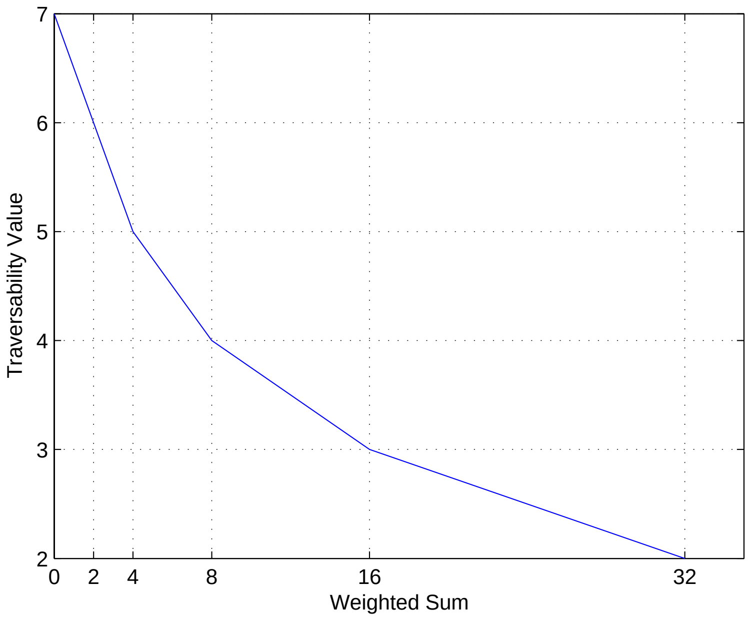


Figure 5-7. Traversability value mapping.

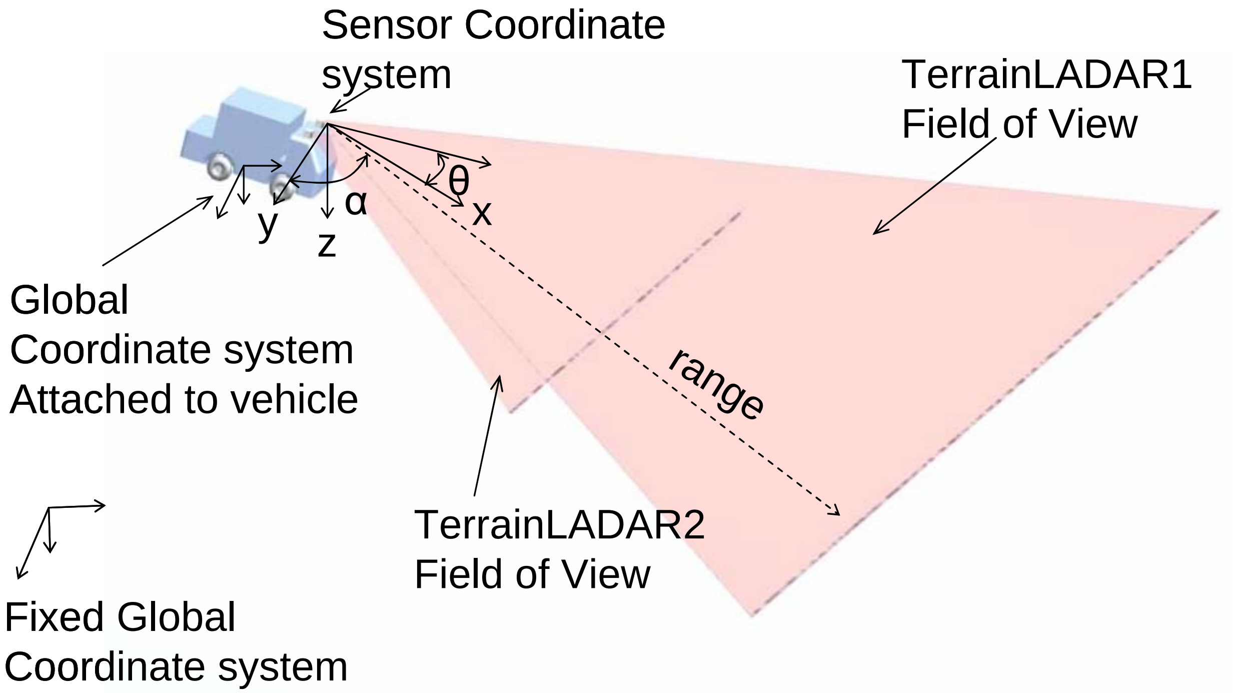

Sensor Coordinate system
TerrainLADAR1 Field of View
θ
x
α
y
z
Global Coordinate system Attached to vehicle
range
TerrainLADAR2 Field of View
Fixed Global Coordinate system

Figure 5-8. Schematic of terrain evaluation sensors.

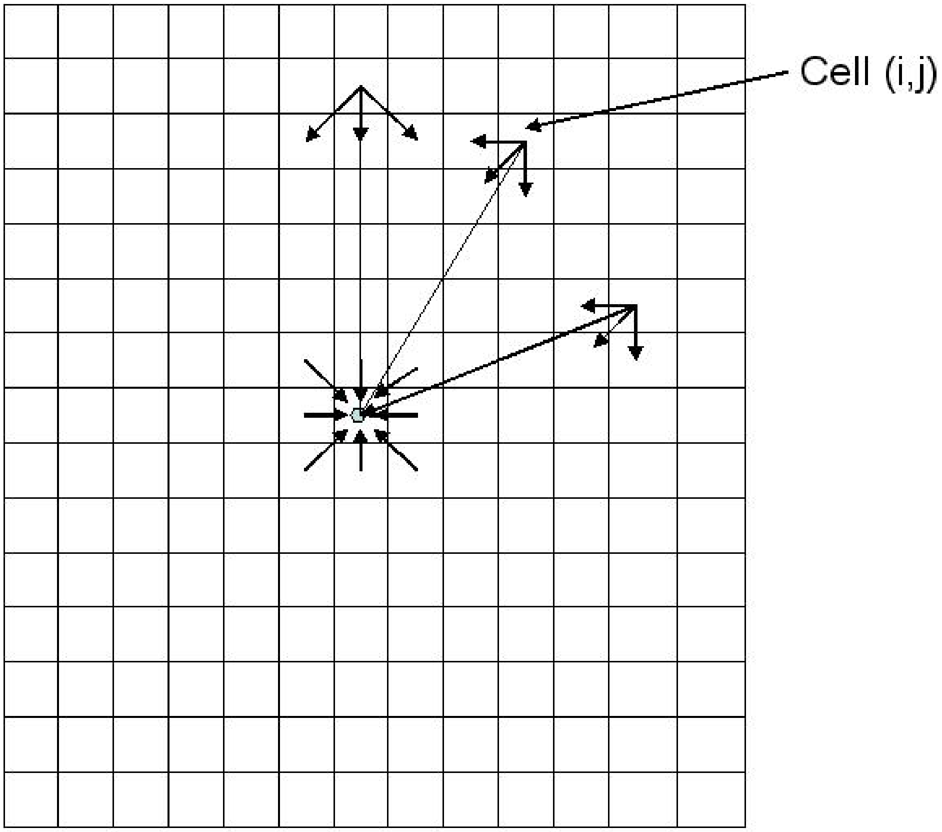


Figure 5-9. Weighted neighborhood analysis

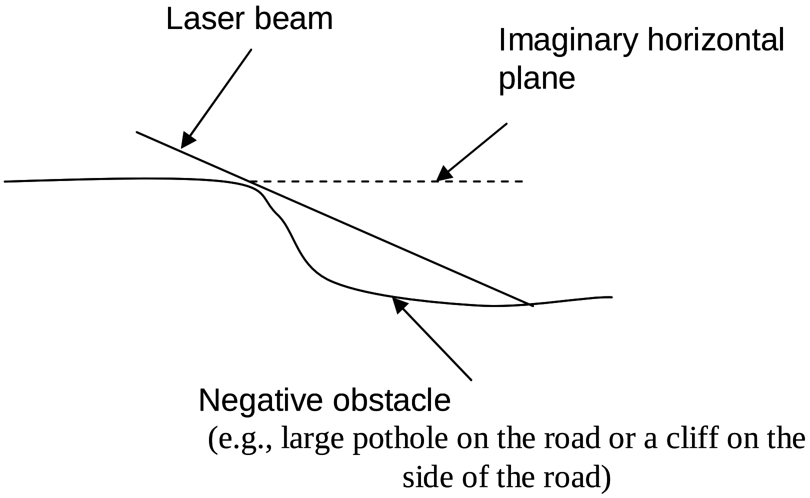

Laser beam
Imaginary horizontal plane
Negative obstacle
(e.g., large pothole on the road or a cliff on the side of the road)

Figure 5-10. Schematic working of negative obstacle detection algorithm.

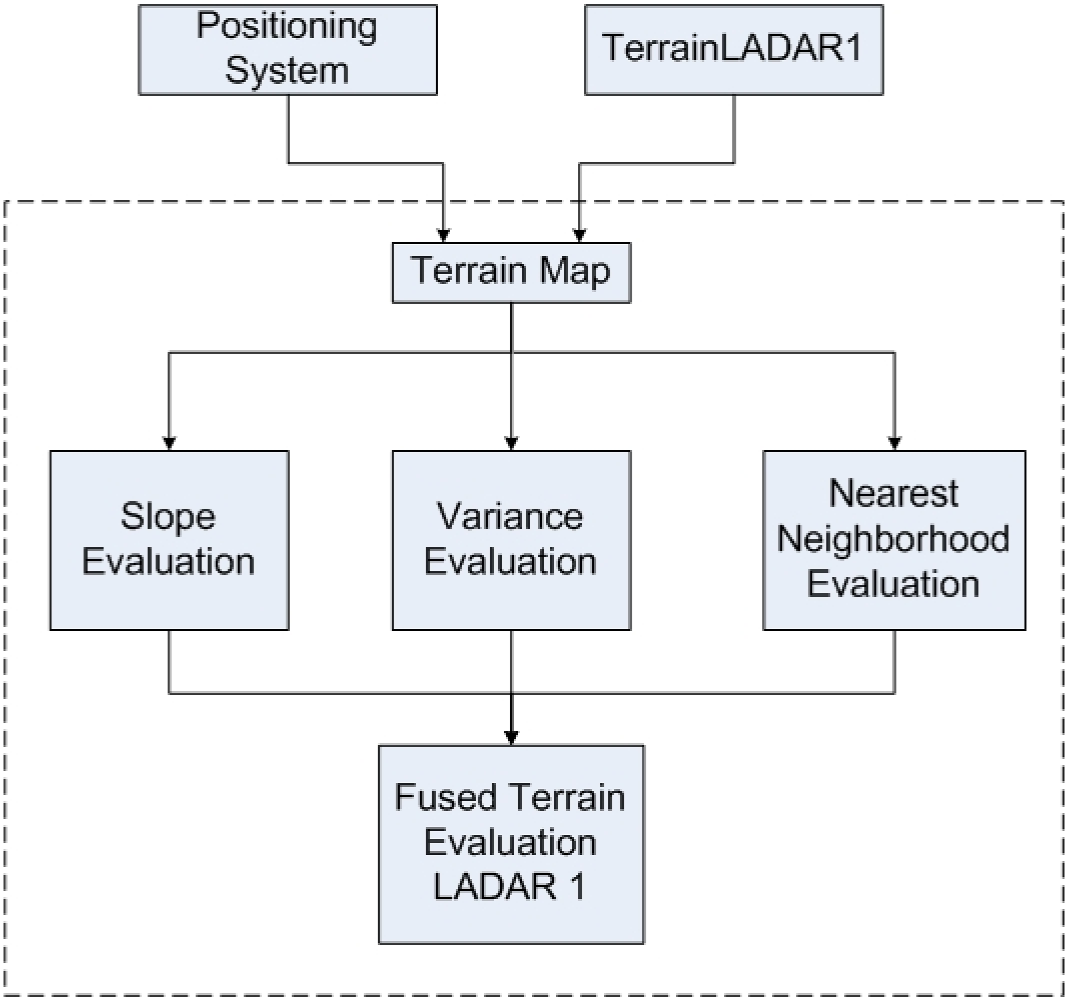


Figure 5-11. Block diagram of terrain evaluation algorithms

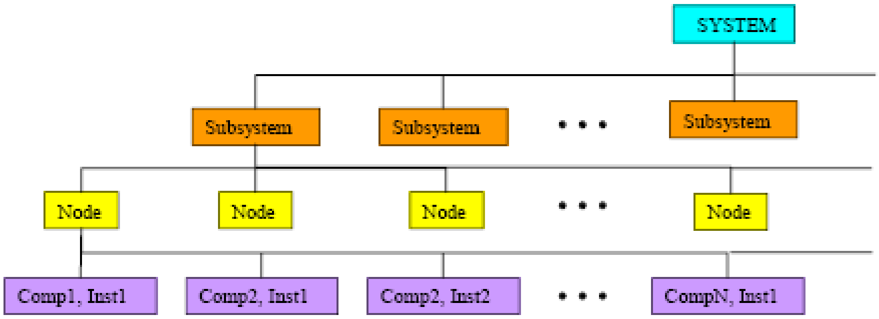


Figure 5-12. JAUS system topology.

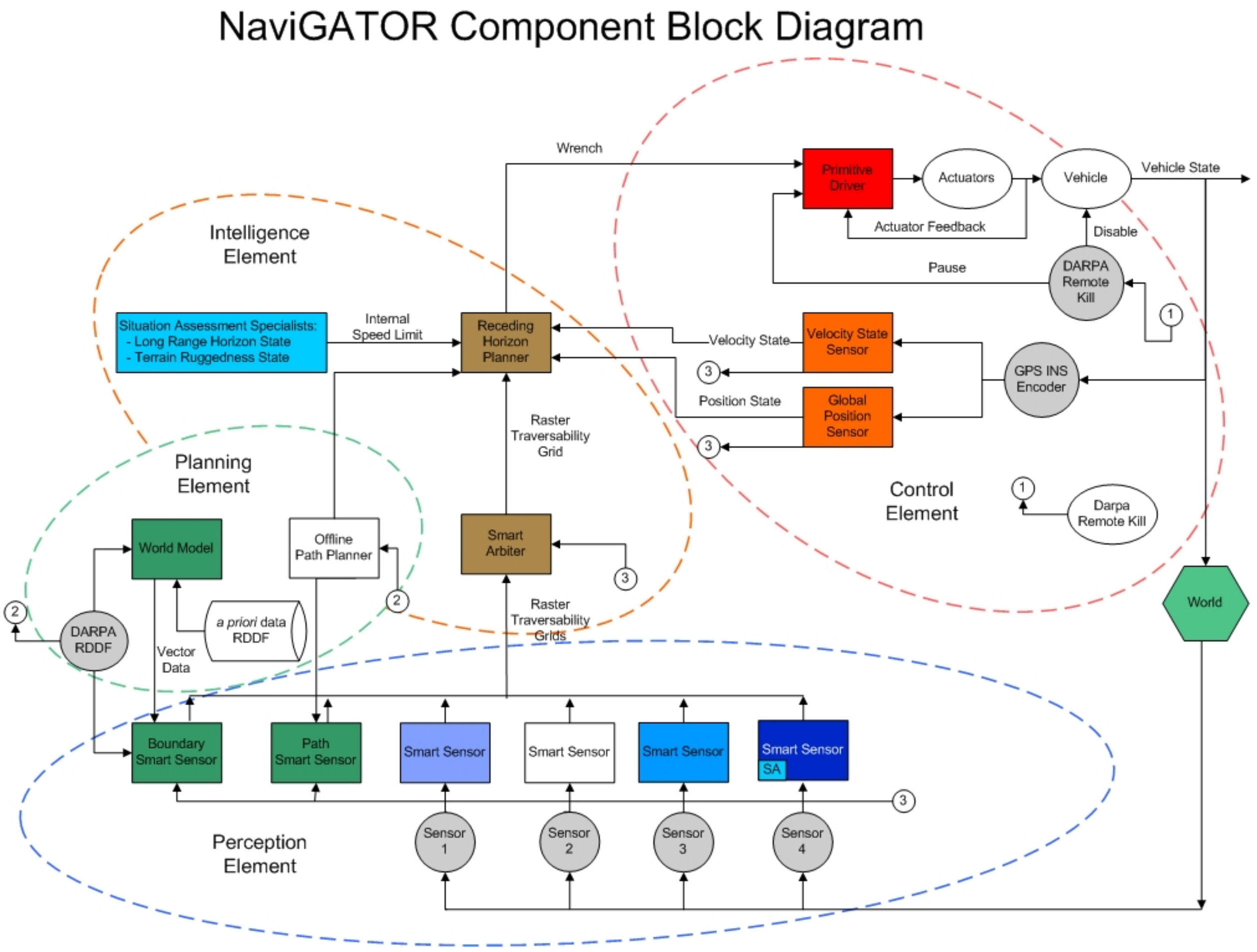


Figure 5-13. JAUS compliant system architecture.

CHAPTER 6
EXPERIMENTAL RESULTS

The developed LADAR Traversability Smart Sensor (LTSS) component was tested on the autonomous platform, NaviGator. In Chapter 3 the NaviGator and the positioning system available on the platform were discussed. The LTSS component is formed from the fusion of two different implementations of the LADAR sensors, the obstacle detection (OD) algorithm and the terrain evaluation (TE) algorithm. Outputs from these two algorithms were fused resulting in the LTSS Traversability Grid. This chapter presents and analyzes the results obtained from the OD sensor, the TE sensor and finally the fusion of these two sensors. The tests in this chapter are conducted at the solar park facility of the University of Florida.

Traversability Grid Visualization

Throughout this chapter the Traversability Grid visualization tool is used to demonstrate the results of the sensor algorithms. The Traversability Grid is represented using the color code shown in Figure 6-1. As shown in the figure a value of 2, which is highly non-traversable is represented by the color red. The color green represents a traversability value of 12, which mean highly favorable to traverse and the color grey is for 7, which is a neutral value. The color shades for the intermediate values are shown in Figure 6-1. The color pink represents a value of 14 which is assigned to cells whose traversability value is unknown. As discussed in the previous chapters the vehicle position is always in the center of the grid. In the Traversability Grid results that follow, the position and direction of the vehicle is always represented by an arrow. Similarly the images showing the test set-up also represent the vehicle position and direction as shown by an arrow.

Obstacle Detection Sensor Results

Obstacle Mapping Results

The obstacle detection (OD) sensor detects positive obstacles of height above a threshold value. If the obstacle is smaller then the threshold height, which is 0.6 m in the present implementation, the sensor cannot detect the obstacle. Similarly, the algorithm cannot offer any opinion regarding negative obstacles or good traversable path. Hence the algorithm can report traversability values in the range of 2 to 7 based on the confidence it has on the obstacle. A value of 7 represents free space, which means there is no positive obstacle reported by the OD sensor; however the cell could contain a negative obstacle or could be a rough, uneven terrain which cannot be determined by this sensor.

The performance of the OD sensor is evaluated on the basis of two important factors: the accuracy in mapping the obstacle and the response time in detecting the obstacle. These performance assessments are evaluated at 3 different speeds of the vehicle: 10 mph, 16 mph and 22 mph. Figure 6-2 shows the experimental set-up for one set of readings. The results of the output from the OD sensor are shown in Figure 6-3 and the summary of the mapping results is presented in Table 6-1. The table shows the comparison between the actual physical distance measured between the barrels and the output from the sensor at vehicle speeds of 10 mph, 16 mph and 22 mph. As seen in Table 6-1 the obstacles are mapped accurately within an error of 0.5 m which is equal to the grid resolution. Hence the maximum error is within 1 cell of the grid.

Another set of readings was taken by changing the position of the barrels and repeating the experiments at the three speeds. Figure 6-4 shows the experimental set-up for the second set of readings. The results are presented in Figure 6-5 and Table 6-2. Similar to the first set of readings the error is within 1 grid cell i.e., 0.5 m.

From the above two set of readings it can be seen that the error in the system is within 1 grid cell and the accuracy is independent of the lateral distance of the obstacle from the vehicle path. It was also seen that the error was repeated between the same barrels in all the three speeds. Since the actual physical distance was measured by a measuring tape, this also causes some error in the measurement of the distance.

Obstacle Detection Response Time

The response time of the sensor algorithm is computed by measuring the distance between the obstacle and the vehicle, at the time when the obstacle is detected. This distance is measured for each of the barrels in the two set of readings shown in Figure 6-2 and Figure 6-4 respectively. A comprehensive summary of the results is presented in Table 6-3. The table shows the distance range in which the obstacles were detected in the direction of the vehicle path. The results of the reading 1 (represented by Figure 6-2) are presented in the movie links in the Objects 6-1, 6-2 and 6-3 for the vehicle speeds of 10, 16 and 22 mph respectively.

Object 6-1. OD reading 1 at 10 mph [ODBarrelTest1_10mph.avi, 100892 KB].

Object 6-2. OD reading 1 at 16 mph [ODBarrelTest1_16mph.avi, 83265 KB].

Object 6-3. OD reading 1 at 22 mph [ODBarrelTest1_22mph.avi, 76214 KB].

Although the OD sensor detects obstacles above a threshold value, it was observed that the height of the obstacle played an important role in how fast the obstacle is detected. The experiment of detecting the obstacle is repeated by increasing the height of the obstacle. Figure 6-6 shows the experimental set-up for the increased height by placing the barrels one on top of the other. The result is presented in the movies in Objects 6-4, 6-5 and 6-6 for the vehicle speeds of 10, 16 and 22 mph.

Object 6-4. OD with increased barrel height at 10 mph [ODBarrelTest2_10mph.avi, 92079 KB].

Object 6-5. OD with increased barrel height at 16 mph [ODBarrelTest2_16mph.avi, 83265 KB].

Object 6-6. OD with increased barrel height at 22 mph [ODBarrelTest2_22mph.avi, 75392 KB].

The analysis of the result for obstacles with increased height is presented in Table 6-4. It can be seen from the results presented in Table 6-3 and Table 6-4, that the distance from which the obstacle is detected from the vehicle is very sensitive to the height of the obstacle, especially when the height of the obstacle is below 2 m. This is because even a small change in the vehicle pitch and roll angles cause a change in the angle of the laser beam. Due to this change the laser beam is no longer horizontal to the ground. For example a change in the pitch angle of 2° would cause a laser beam to shoot at a height difference of 1 m above the critical height at a distance of 30 m from the laser. At high speeds on rough path (such as the environment shown in the above experiments) the roll and pitch changes and the rate of these changes are very high, and hence there is a difference in the performance based on the height of the obstacle.

The above experiments demonstrated the performance of the OD algorithm. The next section presents the results for the TE algorithm and the fusion of the two algorithms.

Fusion of the Obstacle Detection and Terrain Evaluation algorithms

The main task of the terrain evaluation (TE) algorithm is to detect a smooth path and distinguish it from the surroundings. Unlike the obstacle detection (OD) algorithm which can report traversability values only in the range of 2 to 7, the TE algorithm can report values from 2 to 12. As discussed in Chapter 5, the traversability value is computed for each cell based on a set of features. To assess the performance of the TE algorithm and subsequently the result of combining the outputs from the TE and OD sensors, the vehicle is driven on a small paved road within the solar park facility. The actual path is shown in the movie in Object 6-7.

Object 6-7. Test Path [VideoForwardPath.avi, 335689 KB].

The vehicle speed was maintained at approximately10 mph. The same path was driven three times and each time the LTSS component was executed on this path in a different output mode.

The output from the OD, TE and the fused LTSS grid is shown in the movies linked to Objects 6-8, 6-9 and 6-10 respectively.

Object 6-8. OD result for test path [ODFusionTest1_10mph.avi, 356136 KB].

Object 6-9. TE result for test path [TEFusionTest1_10mph.avi, 393154 KB].

Object 6-10. LTSS result for test path [LTSSTest1_10mph.avi, 337804 KB].

For a second set of experiment the same path was driven in the opposite direction. The actual path is shown in the movie linked to Object 11.

Object 6-11. Return test path [VideoReturnPath.avi, 239561 KB].

The output result from the OD algorithm, TE algorithm and the fused LTSS grid are shown in the movies linked to Objects 6-12, 6-13 and 6-14 respectively.

Object 6-12. OD result for return path at 10mph [ODFusionTest2_10mph.avi, 306897 KB].

Object 6-13. TE result for return path at 10mph [TEFusionTest2_10mph.avi, 316651 KB].

Object 6-14. LTSS result for return path at 10mph [LTSSTest2_10mph.avi, 239208 KB].

A couple of scenarios from the above two experiments are selected and discussed here.

Scene 1

The scene 1 is selected from the first of the above two experimental readings. Figure 6-7 shows the image of the environment with barrels, poles, trailers and some name boards. Figure 6-8 shows the outputs from the OD and TE algorithms. As seen in the results the obstacles which can be clearly distinguished from the surroundings mainly due to their height (these include the barrels, name boards, poles and trailers) are mapped very well by the OD sensor. However, the rest of the region is shown as free space without any indication of a favorable traversable path. The TE algorithm makes a good attempt to distinguish the smooth path from the surrounding grass region and also shows the discontinuity at the edge of the path. The classification results from the TE algorithm are based on the absolute scale of traversability defined for each of the features; the slope, variance and the nearest neighbor. As it can be seen although the path is distinguishable, the traversability values for most of the region ranges between 6 and 12. This is because, even though there is a clearly distinguishable paved path, the region around the path is still considered to be drivable by the vehicle. The result after combining the two algorithms is depicted in Figure 6-9. As seen in the figure, the LTSS component takes the advantages of both the algorithms to represent clearly distinguishable obstacles and to follow a smooth path.

Scene 2

The scene 2 is also selected from the first experiment. Figure 6-10, shows a snapshot of the video representing the scene. The results from the individual sensor algorithms are shown in Figure 6-11 and the fused output is shown in Figure 6-12. Similar to scene 1, it can be seen that the fused output has the advantages from both the sensor algorithms and makes a more complete representation of the environment then the output obtained from either one of the algorithm.

Scene 3

The scene 3 is selected from the second set of experiments (i.e., when the vehicle is on its way back). Figure 6-13 shows the environment. The results are presented in Figures 6-14 and 6-15. It can be seen from Figure 6-14 that the OD algorithm identifies and maps obstacles such as the dumpster and the TE algorithm distinguishes the paved path from the surrounding region. The fused output shows a very good representation of the environment.

High Speed Test of the LTSS component

A part of the first experimental path was driven at a higher speed and the results were assessed. The vehicle reached the speed of approximately 22 mph. The vehicle was driven three times to obtain the outputs from the OD sensor , TE sensor, and the fused LTSS Traversability Grid. Objects 6-15, 6-16 and 6-17 shows the output from OD, TE and the fused LTSS algorithm respectively.

Object 6-15. Obstacle detection result at 22mph [ODFusionTest_22mph.avi, 109706 KB].

Object 6-16. Terrain evaluation result at 22mph [TEFusionTest_22mph.avi, 110646 KB].

Object 6-17. LTSS result at 22mph [LTSSTest_22mph.avi, 135207 KB].

The main limitation to achieve better results at higher speeds is the laser update rate. The terrain sensors operate at 18 Hz and at this rate each individual grid cell barely manages to get at the most a single laser scan data above speeds of 20 mph. The speed limitation factor based on the laser update rate is discussed in Chapter 7. Another important factor is the vehicle position and orientation. At higher speeds it is critical to correlate the laser data and the vehicle position fairly accurate to obtain a reasonable point cloud of the laser data. In spite of the above limitations, the developed LTSS component produces fairly good results up to speeds of 20 mph.

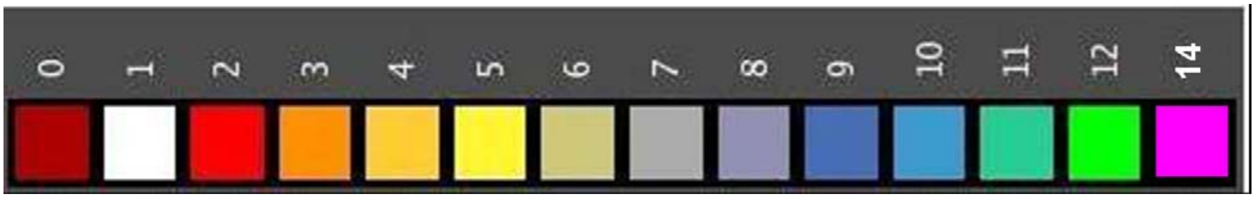


Figure 6-1. Traversability Grid color code.

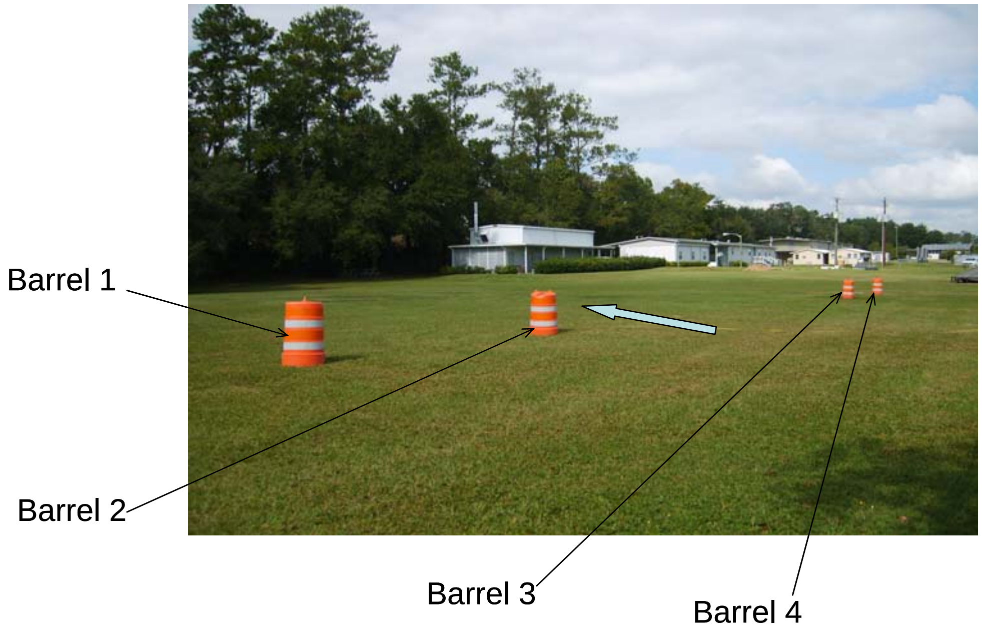


Figure 6-2. Obstacle detection reading 1 experimental set-up.

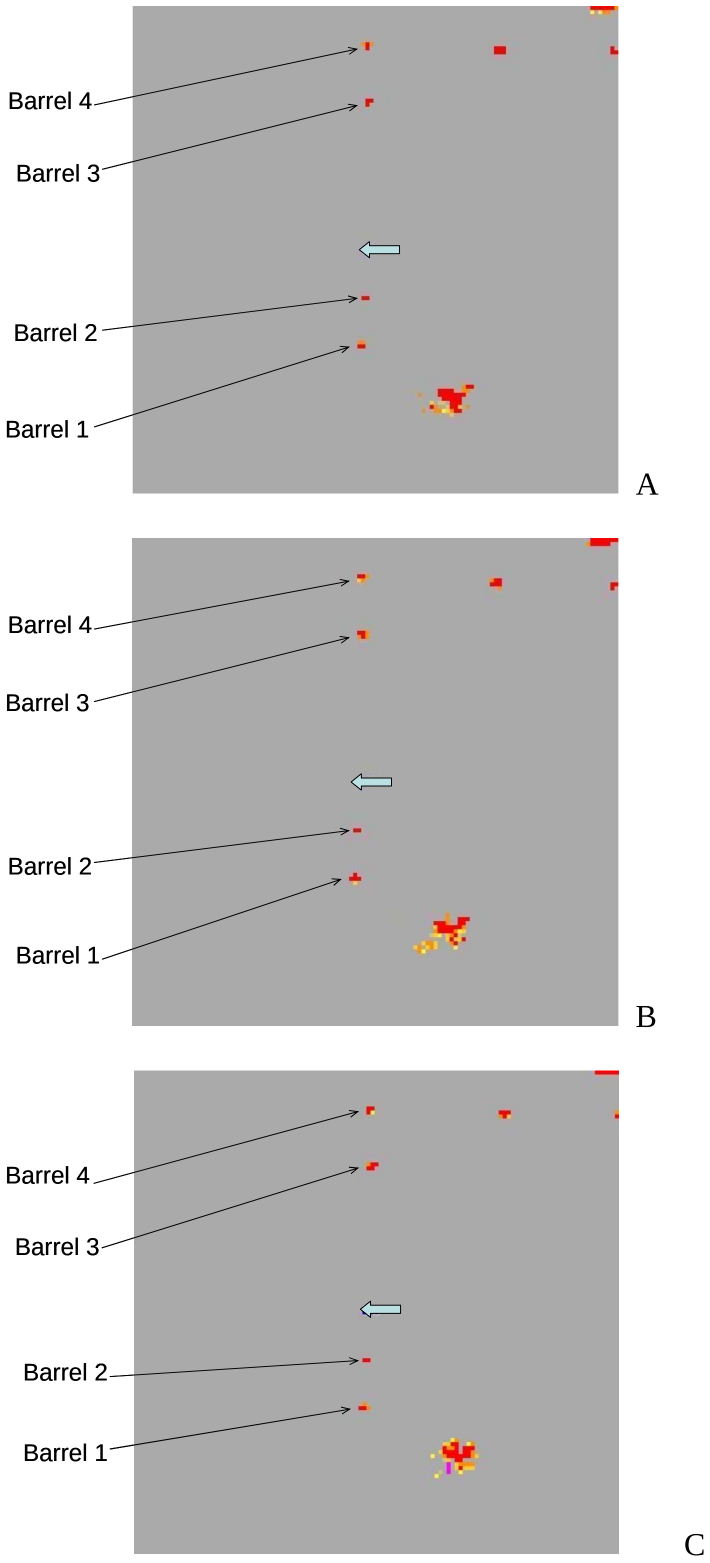


Figure 6-3. Obstacle detection reading 1 output at varied speeds. A) 10 mph B) 16 mph C) 22 mph

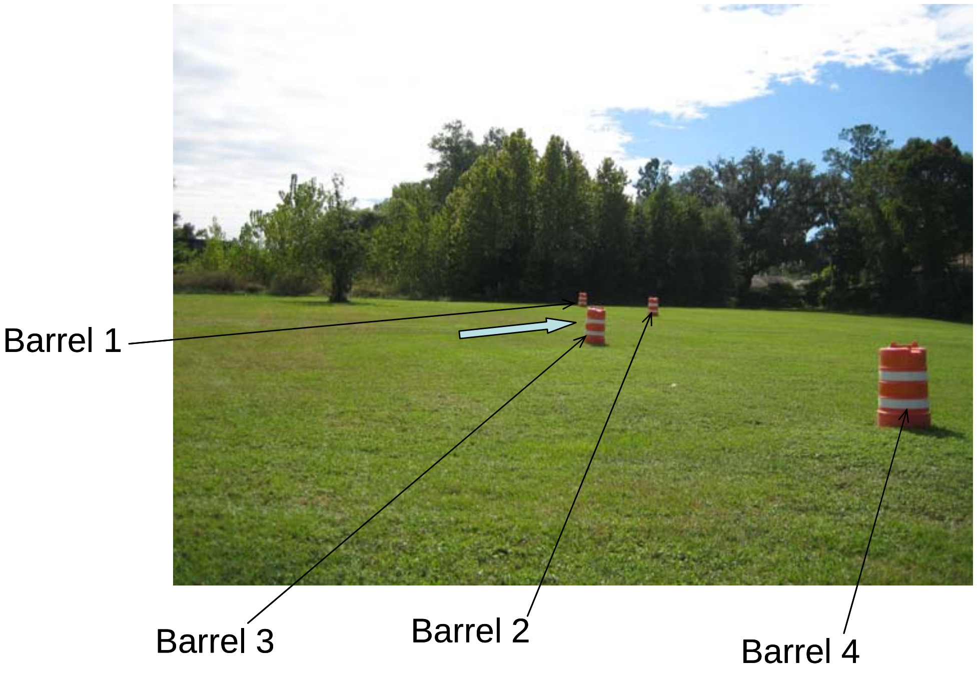


Figure 6-4. Obstacle detection reading 2 experimental set-up.

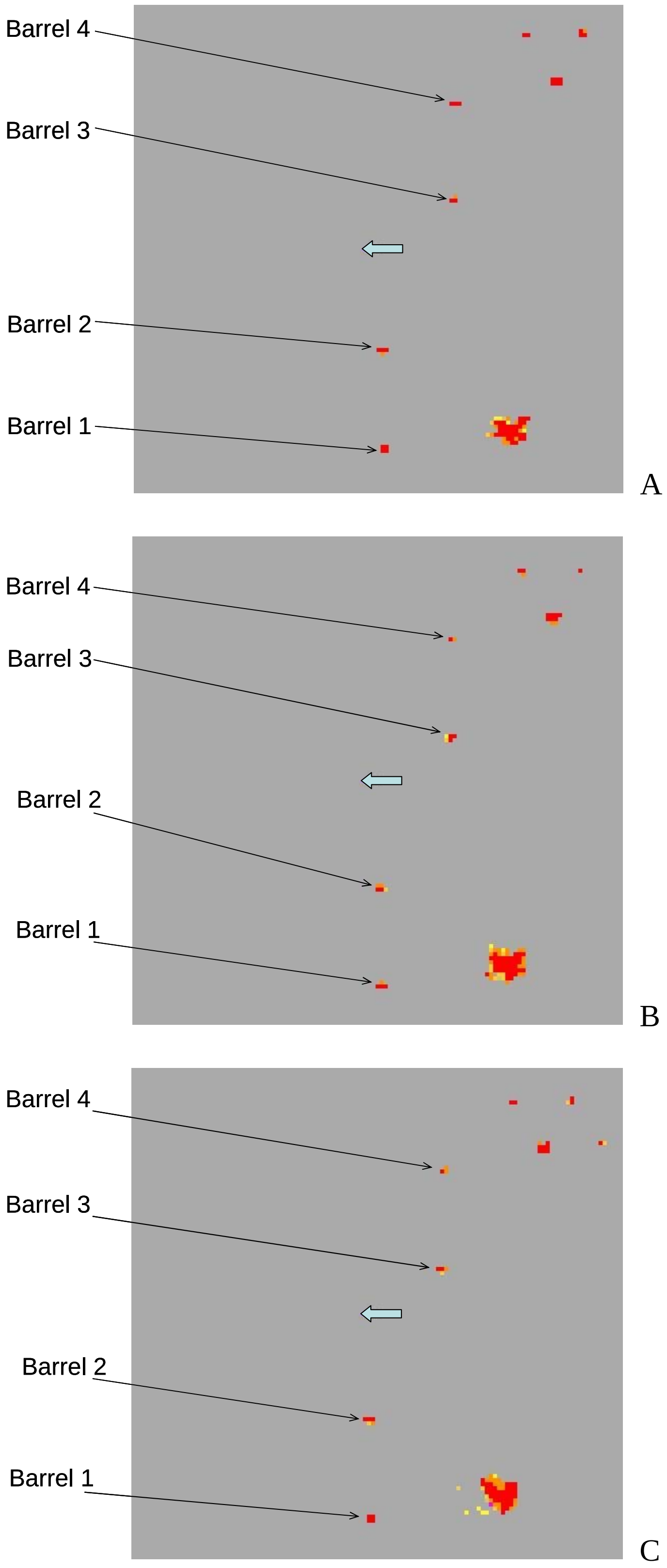


Figure 6-5. Obstacle detection reading 2 output at varied speeds. A) 10 mph. B) 16 mph. C) 22 mph.

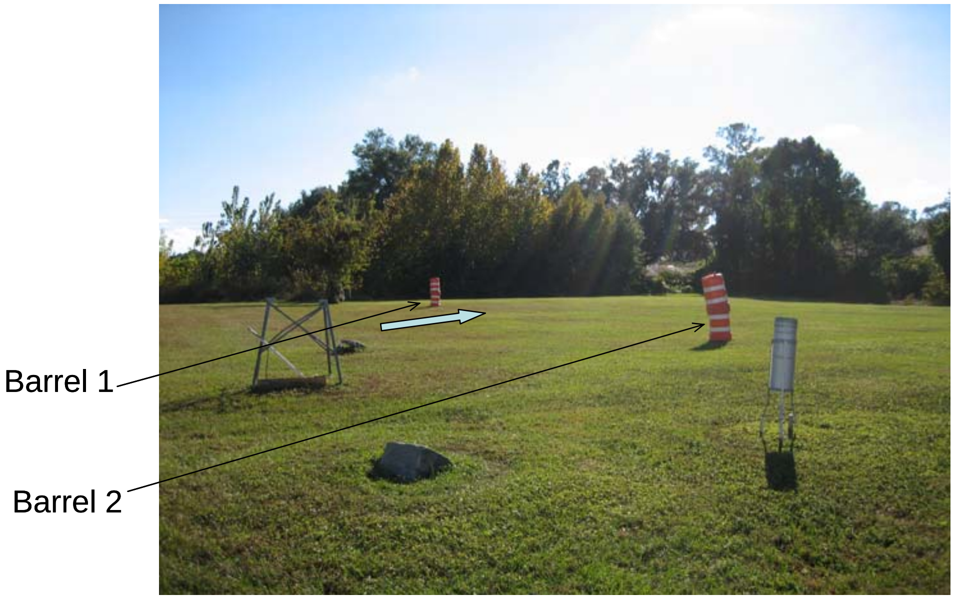


Figure 6-6. Obstacle detection response time with increased height.

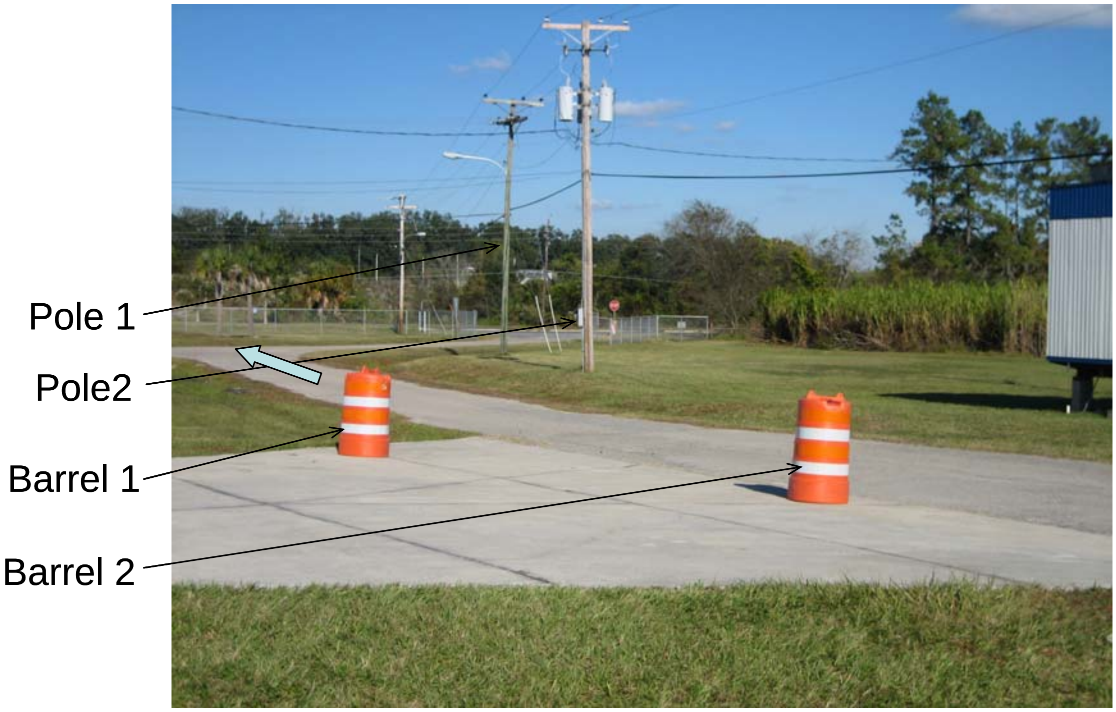


Figure 6-7. Test environment showing scene 1.

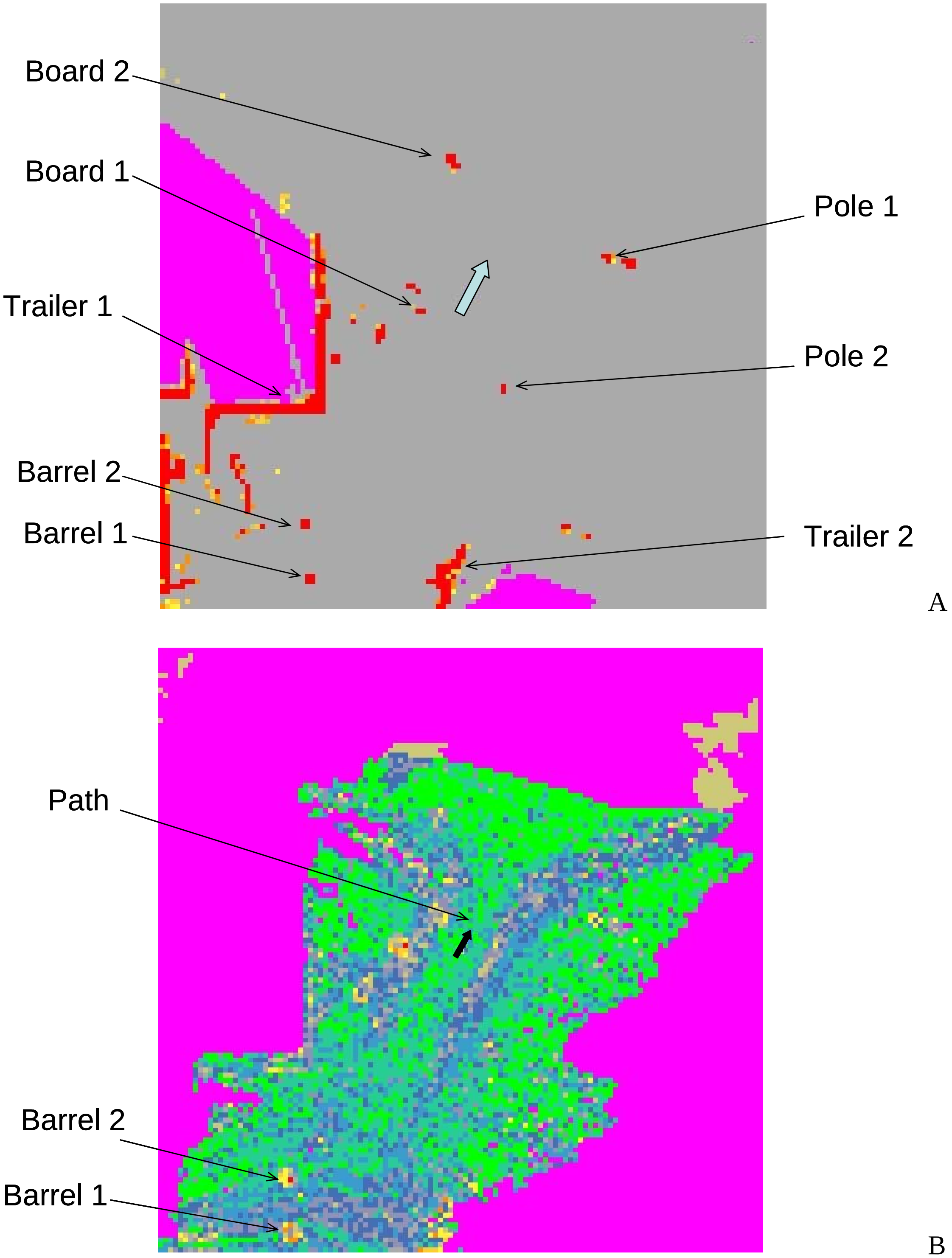


Figure 6-8. Output results for scene 1. A) OD algorithm B) TE algorithm

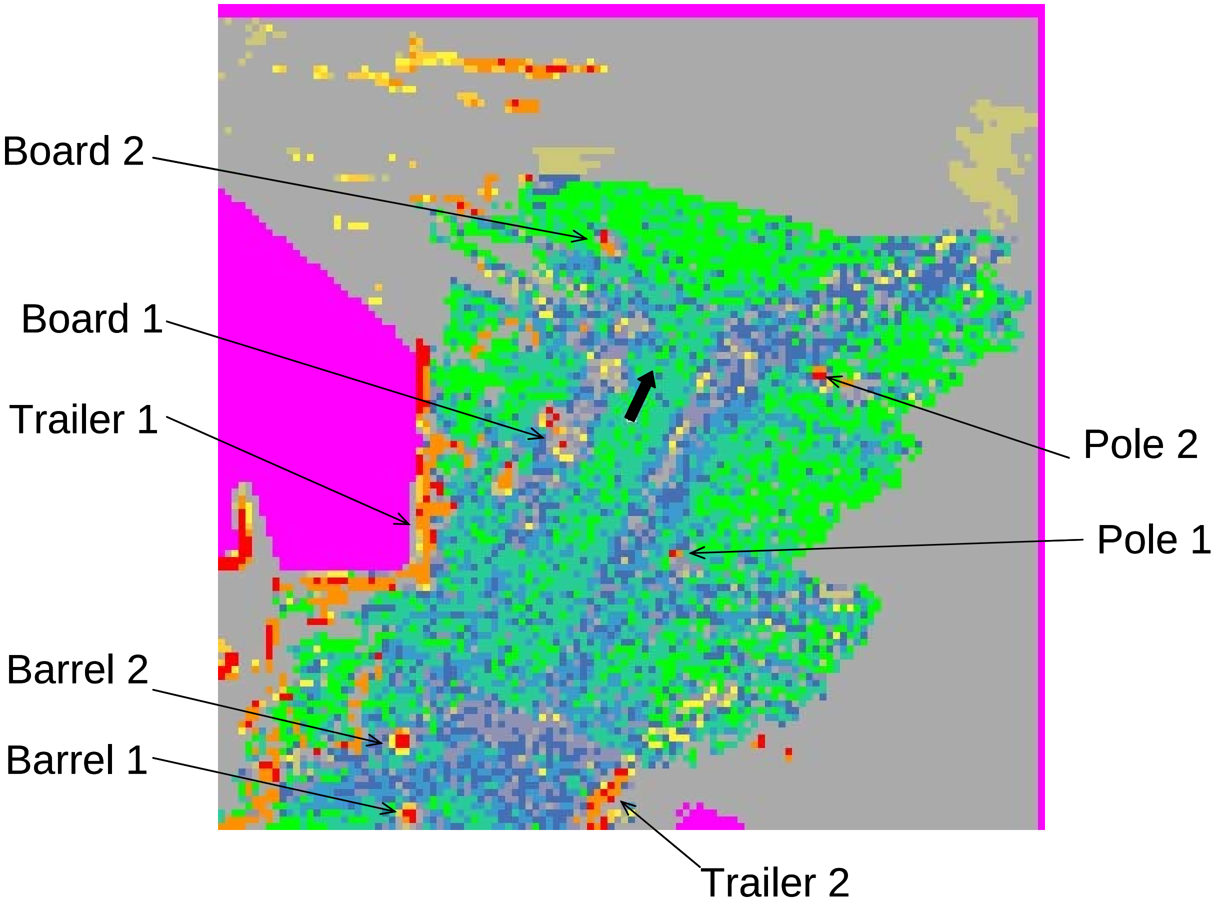


Figure 6-9. Scene 1 LTSS component output environment representation.

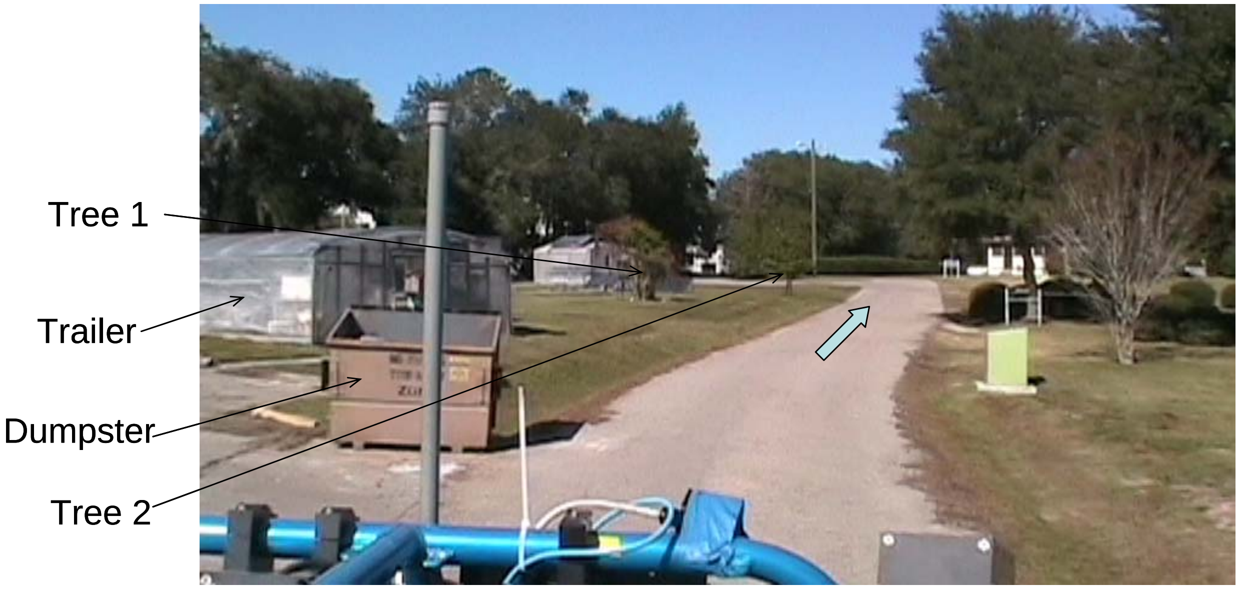


Figure 6-10. Test environment showing scene 2.

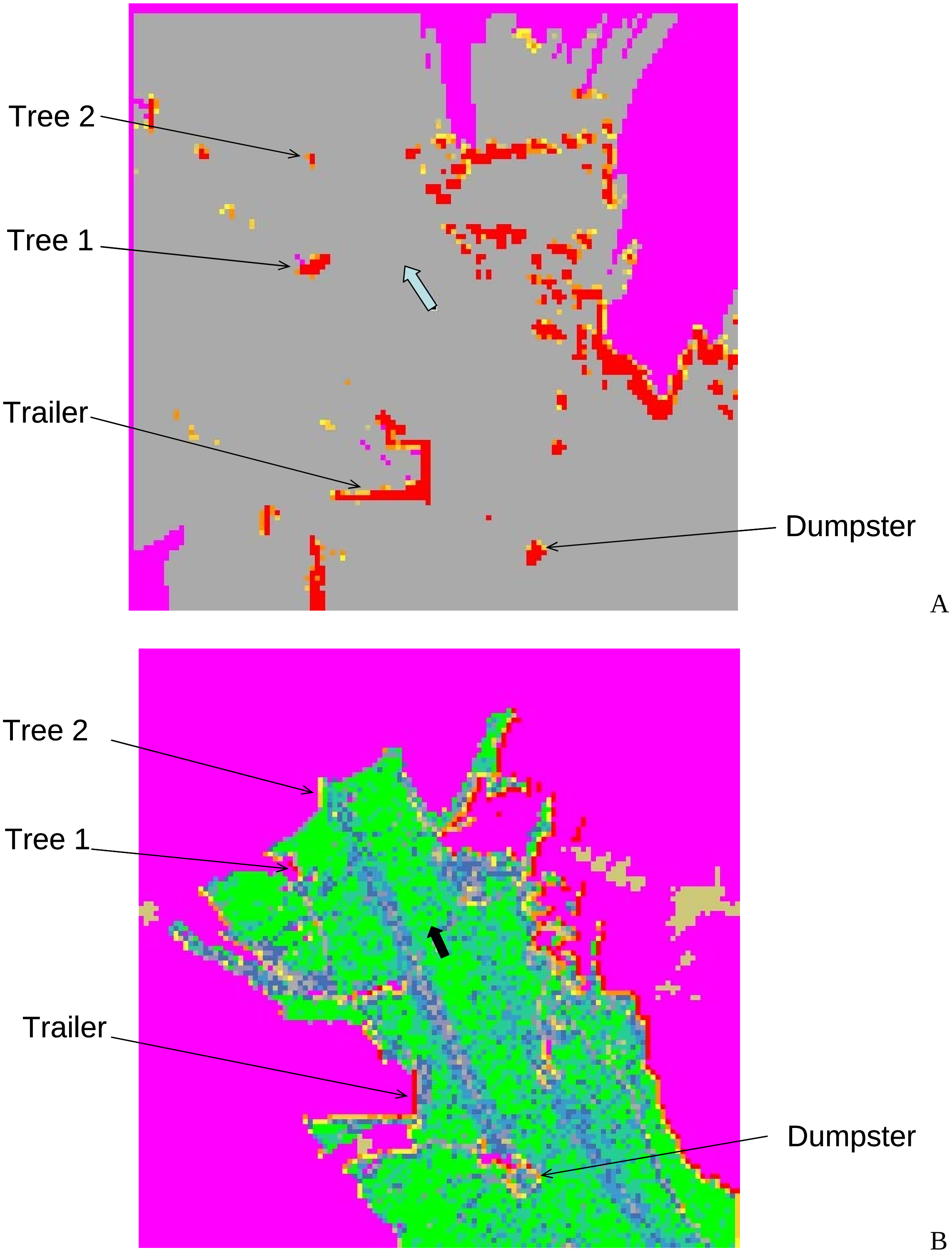


Figure 6-11. Output results for scene 2. A) OD algorithm B) TE algorithm

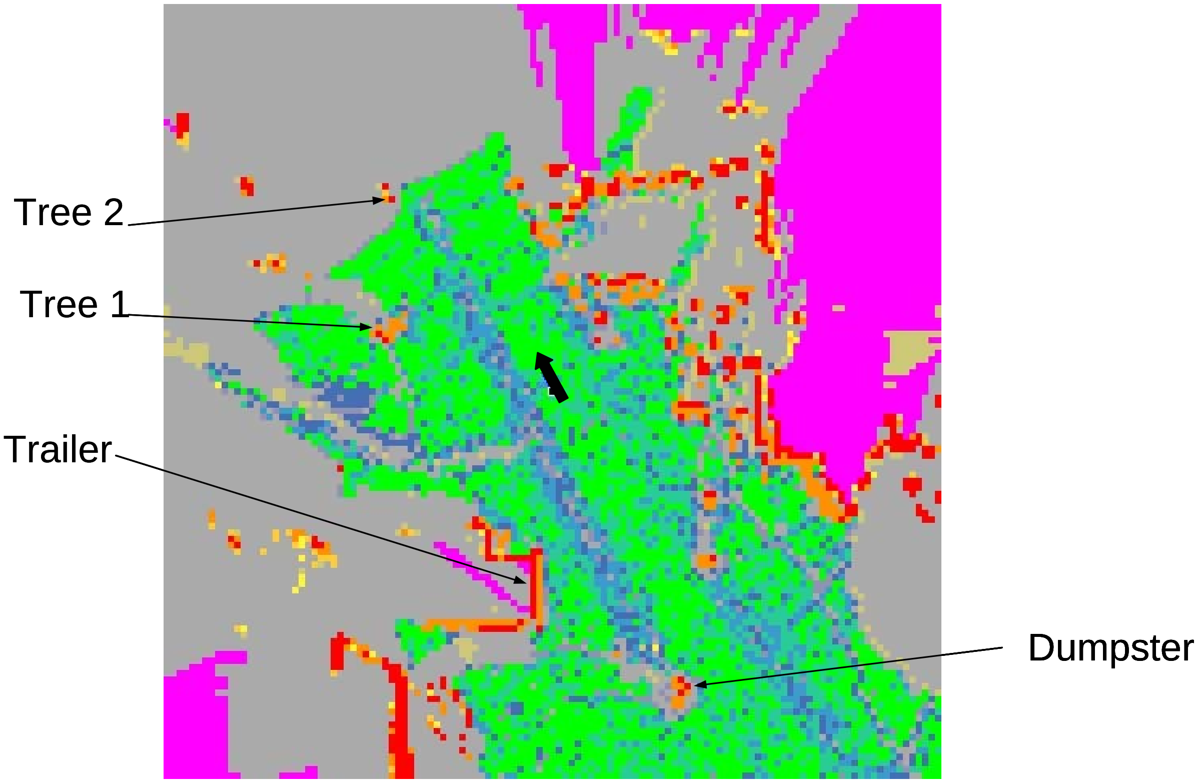


Figure 6-12. Scene 2 LTSS component output environment representation.

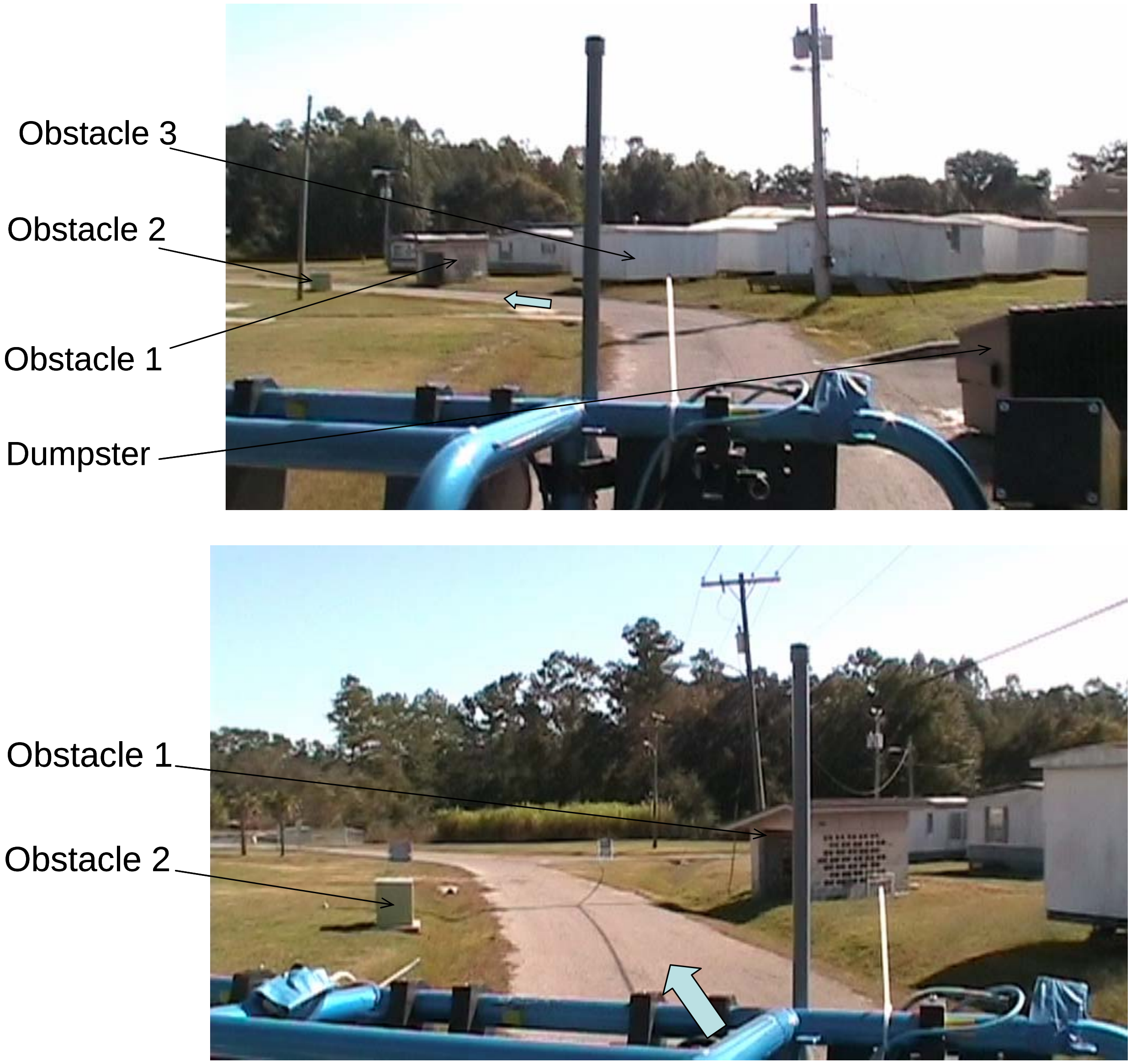


Figure 6-13. Test environment showing scene 3.

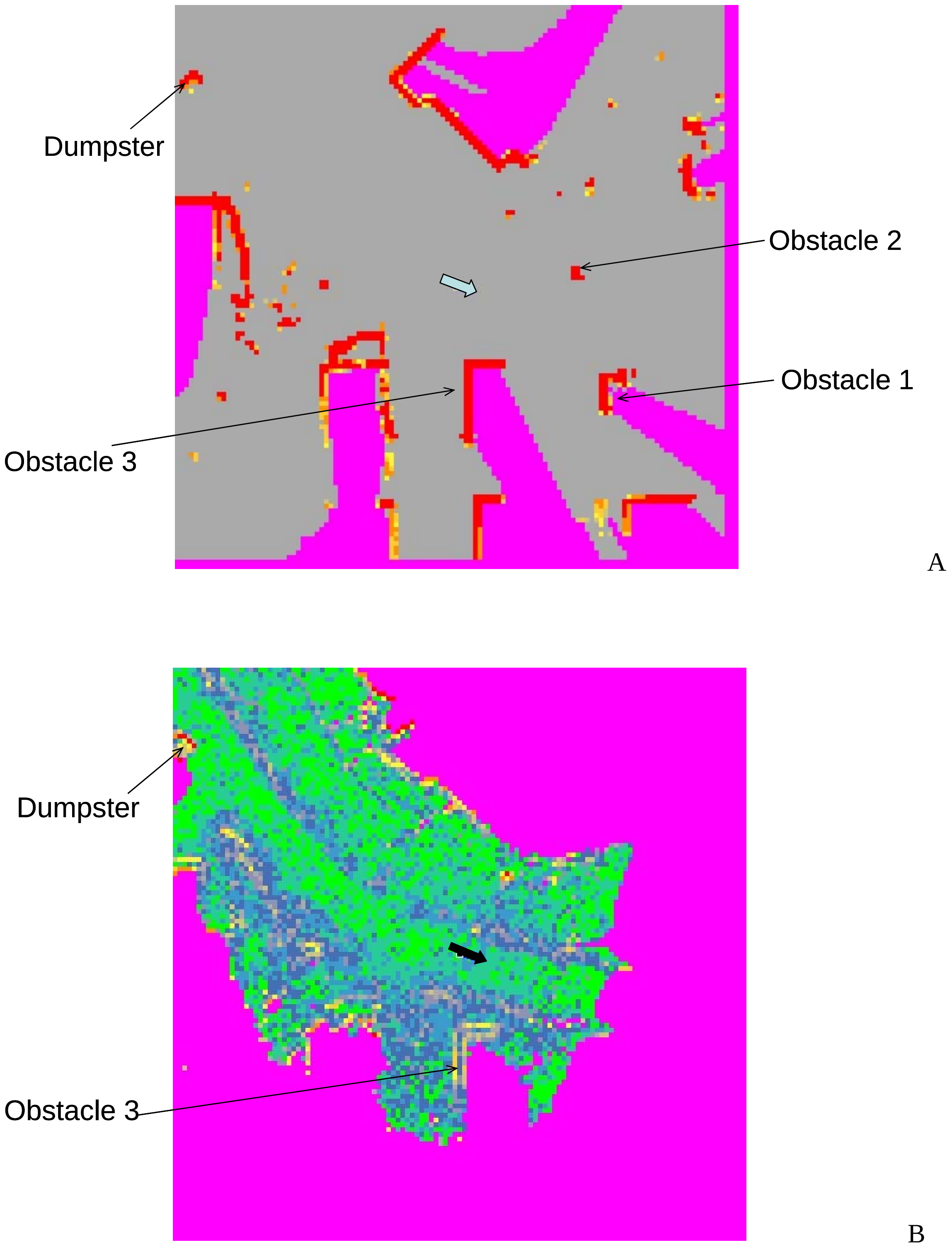


Figure 6-14. Output results for scene 3. A) OD algorithm B) TE algorithm

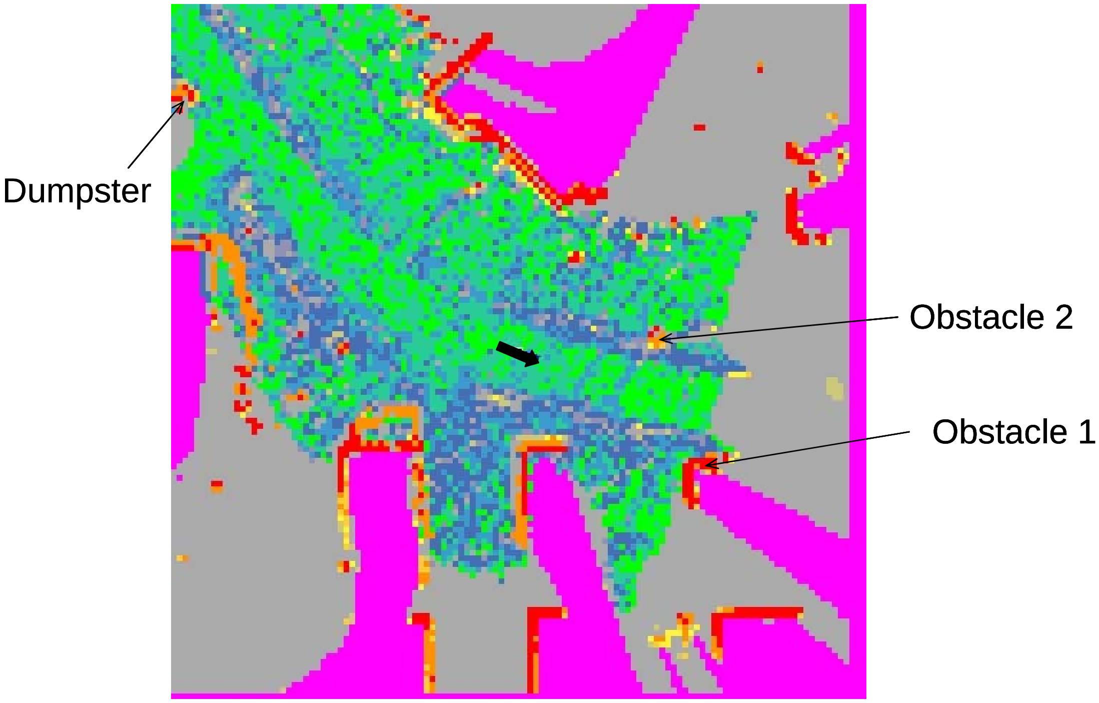


Figure 6-15. Scene 3 LTSS component output environment representation.

Table 6-1. Traversability Grid mapping of obstacle detection algorithm for reading 1.

Distance measured between	Actual distance (m)	Output measurement (m) at 10mph	Output measurement (m) at 16mph	Output measurement (m) at 22mph
Barrel 1 and 2	6	6	6	6
Barrel 2 and 3	24	24	24	24
Barrel 3 and 4	6	6.5	6.5	6.5

Table 6-2. Traversability Grid mapping of obstacle detection algorithm for reading 2.

	Actual Distance (m)	Output measurement (m) at 10mph	Output measurement (m) at 16mph	Output measurement (m) at 22mph
Barrel 1 and 2	12	12	12	12
Barrel 2 and 3	18	18.5	18.5	18.5
Barrel 3 and 4	12	12	12	12

Table 6-3. Response time reading 1

Speed of the vehicle (mph)	Distance at which obstacle first detected (i.e. traversability value < 7) (m)	Distance at which traversability value = 2 (m)
10	29-30	24-25
16	28-30	20-22
22	21-22	14-16

Table 6-4. Response time reading 2 with increased obstacle height

Speed of the vehicle (mph)	Distance at which obstacle first detected (i.e. Traversability value < 7) (m)	Distance at which Traversability value = 2 (m)
10	30	29-30
16	29-30	26-27
22	26-28	20-21

CHAPTER 7
GENERALIZED SENSOR COMPONENT

The previous chapters concentrated on the implementation of the LTSS component on a specific vehicle. Most of the laser based real-time terrain evaluation and obstacle detection implementations discussed in the literature are also vehicle specific and/or terrain specific (i.e., an obstacle detection sensor or a terrain evaluation sensor is developed for a specific vehicle and environmental conditions). To enable a wider use of the developed sensor component, this chapter presents a general guideline to select specific sensor related parameters for implementing the proposed sensor component algorithms on different vehicles or vice a versa given the sensor configuration parameters, what would be the limiting conditions of operation for the proposed algorithms to work.

Figure 7-1 shows the schematic of the overall implementation of sensor component. If the current implementation is examined in a broader view, it can be seen that the obstacle detection algorithm and the negative obstacle detection algorithm discussed in Chapter 5, are the limiting implementations of the terrain evaluation algorithms (slope, variance and neighborhood analysis). While the terrain evaluation algorithms evaluates the terrain characteristic in front of the vehicle from a point cloud of the terrain, the obstacle detection makes sure that there is nothing in front of the vehicle over which it cannot drive and the negative obstacle detection makes sure that there is some surface on which to drive. There are a number of sensor related parameters that would depend on the vehicle on which these sensors are implemented. Some of these sensor related parameters are:

1. Placement of each of the sensor hardware on the vehicle.
2. Sensor tilt angles towards the ground.
3. Sensor field of view.

4. Sensor resolution.
5. Update rate of the sensor.

These sensor parameters are functions of the vehicle specifications and the performance requirements of the vehicle. A list of these specifications would include:

1. Vehicle overall dimensions.
2. Turning radius of the vehicle.
3. Speed range of the vehicle.

The sensor parameters would also be affected by the terrain conditions which could be analyzed in the following terms:

1. Roughness of the road.
2. Maximum expected curvature in the terrain.
3. Presence and magnitude of negative obstacles.
4. Minimum height of moving obstacles expected in the terrain.

The problem statement could thus be stated as, given the vehicle specifications and the performance requirements of the vehicle devise a generalized method to define the actual hardware requirements of the laser sensors and the positioning of the sensors on the vehicle to implement the proposed algorithms.

General Parameters for Obstacle Detection sensor

For the obstacle detection (OD) sensor, the height at which the sensor is mounted is very important. Since this height acts as a threshold value, obstacles below this height will not be detected by the OD sensor; at the same time selecting to place the laser at a lower height would increase the chances of false hits due to ground noise. Ground noise is more prominent in paths with significant up and down hills. Hence the mounting height of the laser depends on the terrain condition, if the terrain is relatively flat, choosing to mount the sensor at a lower height will help detect obstacles with lower heights, however for path with uphill and downhill slopes, the sensor will have to be mounted at a relatively higher height to avoid ground noise. The OD sensor is particularly important for clearing moving obstacles. If the moving obstacle is less then the height of the OD sensor, it will not be possible to detect the obstacle and clear it once it has moved. Although the obstacle could be detected by the terrain evaluation sensors, the problem would be to clear this obstacle once it has moved since terrain evaluation algorithms do not clear the obstacle once it is detected at a particular location.

General Parameters for Terrain Evaluation sensor

One of the important vehicle dimensions is the height at which the terrain mapping laser can be mounted on the vehicle. Let *h* be the height at which the laser can be mounted on the vehicle. The other laser mounting parameter is the tilt angle at which the laser is mounted. The angle of tilt is governed by the terrain condition. The tilt angle cannot be greater then the expected slope changes in the terrain. If angle θ is the tilt angle of the laser with respect to the vehicle ground plane, than this angle should be greater then the change in slope, found on the path, otherwise the laser beam readings will not hit the terrain surface. The above two parameters govern the look ahead distance, d_l of the sensor as follows:

$$d_l = \frac{h}{\tan\theta} \tag{7.1}$$

The current implementation used two lasers for terrain evaluation; one with a smaller tilt angle to get a greater look ahead distance, the other one was used with a bigger tilt angle to be able to scan the ground ahead with higher slope changes, but at a distance much closer to the vehicle. The TerrainLADAR1 discussed in Chapter 5 is mounted at an angle of 6° and the TerrainLADAR2 is mounted at an angle of 12°. Figure 7-1 Both the lasers are placed at a height of 1.9m. The look ahead distance for the TerrainLADAR1 is computed as:

$$d_l = \frac{1.9}{\tan 6} = 18.07\ m$$

Similarly the look ahead distance for the TerrainLADAR2 is:

$$d_l = \frac{1.9}{\tan 12} = 8.94\ m$$

Figure 7-1 shows the three parameters for the TerainLADAR1. The look ahead distance is an important digit in deciding the speed of the vehicle. This distance is the time the vehicle has to avoid obstacles and travel a smooth path. But at the same time, the speed of the vehicle is also limited by the update rate of the laser and the expected grid resolution. To evaluate the grid, each cell in the grid should have sufficient amount of data points. To make sure that at least data from one single laser scan are assigned to a cell the following relation has to hold:

$$LaserUpdateRate >= \frac{v}{Grid\ \mathrm{Re}\, solution} \tag{7.2}$$

where, *v* is the speed of the vehicle.

For example in the case of the NaviGator, the update rate of the terrain laser sensors is 18 Hz. and the implemented grid resolution is 0.5 m. Hence to obtain laser data in each cell in the grid within the field of view of the laser the speed of the vehicle has to be limited to:

$$v = LaserUpdateRate * Grid\ \mathrm{Re}\, solution$$
$$v = 18 * 0.5$$
$$v = 9\ m/s$$

When the vehicle is driving on a straight road, the sensor field of view is in the direction of the vehicle travel; however when the vehicle is making a turn this is not the case. While making a turn the sensor field of view is in a direction tangential to the circle defined by the curved path followed by the vehicle. For safe driving, one of the sensor design requirements is to specify a parameter which defines the minimum width of the terrain from either side of the centerline of the vehicle drive path that should be in the sensor field of view at any time. This design parameter, 'w_m', is shown in Figure 7-2. To satisfy this design condition, the vehicle drive path is limited to a minimum allowable radius of curvature, 'R'. The expression for R can be expressed in terms of the parameters defining the sensor field of view. As shown in the figure the sensor field of view is defined by the angle, α, and the look ahead distance, d_l. Consider the triangle OAB in the Figure 7-2, the angle b can be expressed as:

$$b = 90 - \frac{\alpha}{2}\,. \tag{7.3}$$

Using the cosine rule for the triangle OAB, the following expression is obtained:

$$\left(R - \frac{w_m}{2}\right)^2 = R^2 + d_l^{\,2}\sec^2\left(\frac{\alpha}{2}\right) - 2Rd_l\sec\left(\frac{\alpha}{2}\right)\cos\left(90 - \frac{\alpha}{2}\right). \tag{7.4}$$

Solving the Equation 7.4, R is obtained as:

$$R = \frac{d_l^{\,2} * \sec^2\left(\frac{\alpha}{2}\right) - \frac{w_m^{\,2}}{4}}{2d_l\sec\left(\frac{\alpha}{2}\right)\cos\left(90 - \frac{\alpha}{2}\right) - w_m}\,. \tag{7.5}$$

Equation 7.5 expresses R in terms of the sensor parameters and the minimum width, which is a design requirement. In case of the terrain sensor implementation of the NaviGator, consider the TerrainLADAR1 sensor which scans the terrain at a distance of 18 m in front of the vehicle. Let the minimum required width, w_m be 28 m, (i.e. the sensor should scan a distance of at least 14 m on each side of the path center line). Using the above formula,

$$R = \frac{18^2 * \sec^2\left(\frac{100}{2}\right) - \frac{28^2}{4}}{2*18*\sec\left(\frac{100}{2}\right)\cos\left(90 - \frac{100}{2}\right) - 28}$$

$$R = 39.47m.$$

Thus the above condition of scanning the minimum width across the path center line can be achieved only for a radius of curvature above or equal to the computed value.

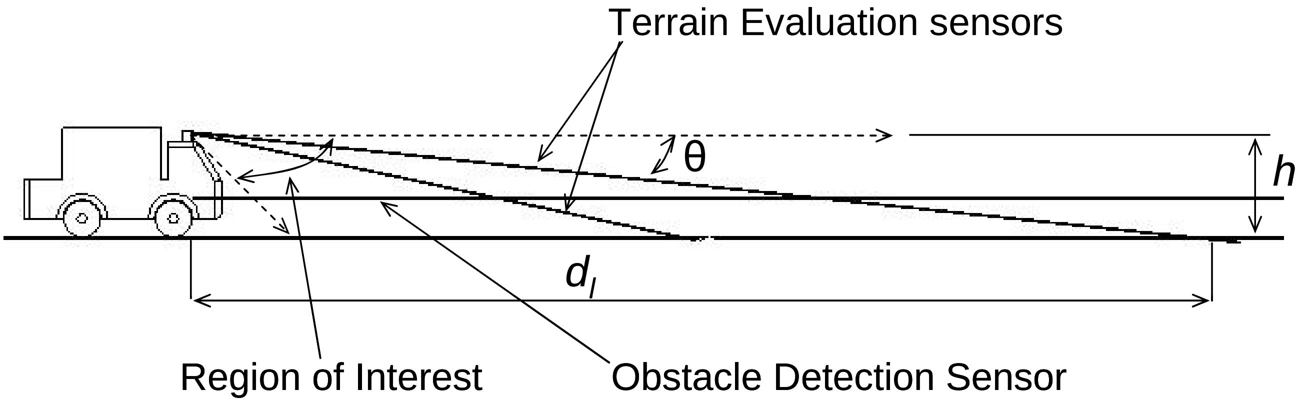


Figure 7-1. Sensor configuration.

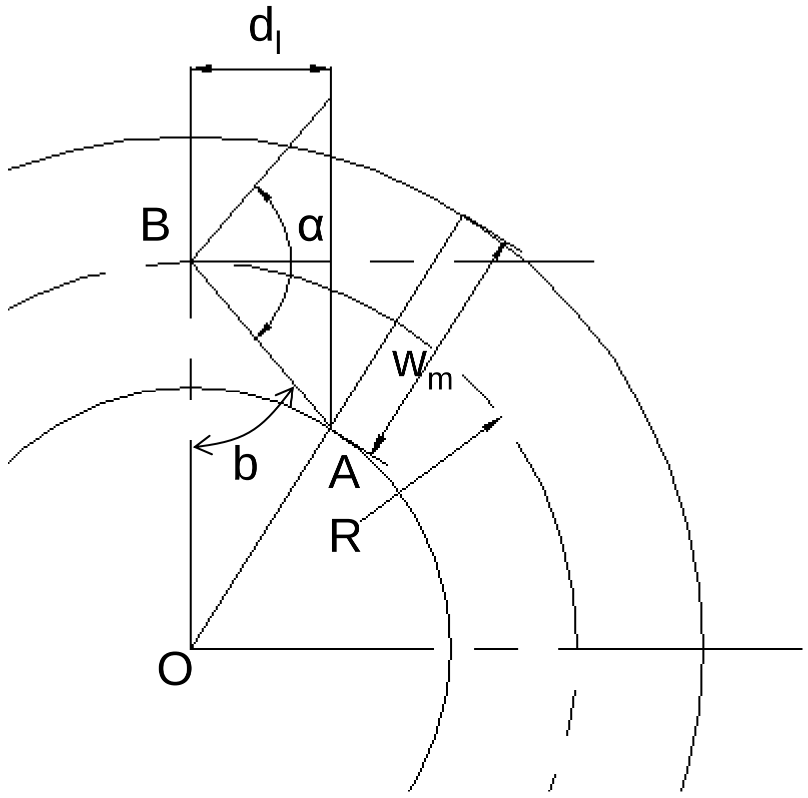


Figure 7-2. Minimum radius of curvature

CHAPTER 8
CONCLUSION AND FUTURE WORK

The research presented an approach to represent the environment around an autonomous vehicle. Two different implementations of laser sensors to evaluate the surroundings were presented. The advantages and disadvantages of each were discussed and a sensor fusion technique to combine the outputs of these two algorithms was presented.

The dissertation presented a novel technique, the weighted neighborhood analysis and the results from this algorithm were fused with the slope and variance algorithms to give an estimate of the terrain surrounding the autonomous vehicle. The results obtained from the terrain evaluation algorithms were very promising.

Instead of using a binary classification of traversable or obstacle, a traversability scale was used to define the environment. The traversability scale allowed representing a wide range of the terrain. It was possible to distinguish between varying degrees of obstacleness. The traversability scale helped the planner to propose an optimal path of travel at every instant of time. The next step is to use a finer traversability scale. A traversability scale of 0:63 instead of the 0:15 will give a better resolution of the environment and hence should be considered for implementation.

The terrain mapping algorithm discussed in the literature does not implement a time history of data points. The only way the latest data are accounted for in the present scheme is by replacing the old data points with the new data points once the cell has reached a maximum allowed number of data points. The problem with this scheme is that if there is a moving obstacle or if the sensor registers an erroneous hit within a cell, there is no way to clear it unless the same region is scanned again by the sensor. A better way to approach this problem would be to time stamp the data points. The time stamped data points can be assigned weights. If the vehicle gets stuck due to an erroneous data point, the confidence on the data point can be lowered with time.

The vehicle orientation was accounted for mapping the data points. The vehicle orientation also provides a very valuable input of the terrain characteristics. The rate of change of the orientation angles provides a good estimate of the roughness of the path the vehicle is traveling on. It was also observed that the magnitude of error in the mapping is related to the rate of change of orientation angles. On a rougher terrain these rates are higher and so are the errors in mapping the data. The final results of the sensor component could be improved by incorporating this information in the rule based scheme.

The grid resolution used for the current implementation was 0.5m. Considering the size of the vehicle and the outdoor environment, the resolution gave good results. However, it would be worth trying to use a higher resolution grid such as a 0.25m and compare the results. It is generally desirable to have a higher resolution grid in the area close to the vehicle and as the distance from the vehicle increases a grid with a lower resolution could serve the purpose. This can be achieved by maintaining the data in two grids with different resolutions; a higher resolution grid with smaller overall dimensions close to the vehicle and another lower resolution grid covering a larger area around the vehicle. A different way to serve the purpose of higher resolution near the center of the grid (vehicle position) but at the same time cover a larger area would be the implementation of a polar grid. A polar grid inherently gives a higher resolution in the region close to the vehicle, and as the distance increases the resolution of the grid decreases.

The certainty factors technique for uncertainty management has wide applications in the medical field. However, the use of certainty factors in sensor fusion is a novel approach. One of the biggest advantages of the scheme is its ability to combine controversial evidences. In future, the possibility of combining outputs from other sensors such as the monocular vision using rule-based implementation of certainty factors should be studied.

The lasers used for mapping the terrain were 2-dimensional range scanners (i.e. they produced a single line scan image) and the map was built by coupling these line scans with the vehicle motion. Although commercial laser scanners are available with a horizontal and vertical field of view, they are highly expensive. A good alternative to generate a multi-line scanner is to provide a tilt mechanism to the single line scanners. The tilt motion produces multiple lines of scans at different angles. The mechanism can either be used to point the laser at the required angle or line scans can be generated at a uniform distance. To take advantage of the tilt motion, the laser scan update rate should be high enough to be able to scan multiple lines at the required speed.

APPENDIX
TRAVERSABILITY VALUE MAPPING FOR TERRAIN EVALUATION ALGORITHMS

The table below shows the mapping of the criterion used to evaluate the terrain in to the traversability values. These tables were implemented as configuration parameters in a config file.

Table A-1. Mapping of slope values to traversability value.

Slope (degrees)	Traversability Value
<= 10	12
> 10 & <= 20	11
> 20 & <= 30	10
> 30 & <= 32	9
> 32 & <= 35	8
> 35 & <= 40	7
>40 & <= 50	6
>50 & <= 60	5
>60 & <=80	4
>80 & <= 85	3
>85 & <= 90	2

Table A-2. Mapping of the variance values in to traversability values.

Variance (m)	Traversability value
<= 0.0002	12
> 0.0002 & <= 0.0003	11
> 0.0003 & <= 0.0004	10
> 0.0004 & <= 0.0005	9
> 0.0005 & <= 0.001	8
> 0.001 & <= 0.003	7
> 0.003 & <= 0.05	6
> 0.05 & <= 0.1	5
> 0.1 & <= 0.2	4
> 0.2 & <= 0.4	3
> 0.4 & <= 1.0	2

Table A-3. Mapping of neighborhood values to traversability value.

Neighborhood value (m)	Traversability value
<= 0.08	12
> 0.08 & <= 0.16	11
> 0.16 & <= 0.2	10
> 0.2 & <= 0.25	9
> 0.25 & <= 0.3	8
> 0.3 & <= 0.35	7
> 0.35 & <= 0.4	6
> 0.4 & <= 0.5	5
> 0.5 & <= 0.6	4
> 0.6 & <= 0.8	3
>0.8 & <= 2.0	2

LIST OF REFERENCES

Asada, M. (1988, November). Map building for a mobile Robot from sensory data. IEEE transactions on Systems, Man and Cybernetics, 20(6), 1326-1336.

Bhatavia, P., & Singh, S. (2002, May). Obstacle detection in smooth high curvature terrain. IEEE International Conference on Robotics and Automation (ICRA 2002), Washington, D.C., 3, 3062-3067.

Braid, D., Broggi, A., & Schmiedel, G. (2006, June). The TerraMax autonomous vehicle concludes the 2005 DARPA Grand Challenge. Intelligent Vehicles Symposium 2006, Tokyo, Japan, 534-539.

Carmer, D., & Peterson, L. (1996, February). Laser radar in robotics. Proceedings of the IEEE, 84(2), 299-320.

Davis, I., Kelly, A., Stentz, A., & Matthies, L. (1995, September). Terrain typing for real robots. Proceedings of IEEE Intelligent Vehicles, 400-405.

Dima, C., Vandapel, N., & Hebert, M. (2003, October). Sensor and classifier fusion for outdoor obstacle detection: an application of data fusion to autonomous off-road navigation. 32nd Applied Imagery Recognition Workshop (AIPR2003), IEEE Computer Society.

Dima, C., Vandapel, N., & Hebert, M., (2004, April). Classifier fusion for outdoor obstacle detection. IEEE International Conference on Robotics and Automation, 1, 665-671.

Foessel-Bunting, A. (2000, November). Radar sensor model for three-dimensional map building. Proc. SPIE, Mobile Robots XV and Telemanipulator and Telepresence Technologies VII, SPIE, 4195.

Foley, J., Dam, A., Feiner, S., & Hughes, J. (1990). Computer Graphics: Principles and Practice, Addison Wesley, Massachusetts.

Goodridge, S., Luo, R., & Kay, M. (1994, October). Multi-layered fuzzy behavior fusion for real-time control of systems with many sensors. IEEE International conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, NV, 272-299.

Gonzalez, A., & Dankel, D. (1993). The engineering of knowledge based systems: theory and practice, Prentice Hall.

Hancock, J., Langer, D., Hebert, M., Sullivan, R., Ingimarson, D., Hoffman, D., Mettenleiter, M., & Froehlich, C. (1998, May). Active laser radar for high-performance measurements. IEEE International Conference on Robotics and Automation, Leuven, Belgium, 2, 1465-1470.

Hancock, J., Hebert, M., & Thorpe, C. (1998, October). Laser intensity-based obstacle detection, IEEE International Conference on Intelligent Robots and Systems, Victoria, BC, Canada, 3, 1541-1546.

Huber, D., & Hebert, M. (1999). A new approach to 3-D terrain mapping. IEEE International Conference on Intelligent Robots and Systems, Kyongju, South Korea, 2, 1121-1127.

JAUS. (2005). Joint architecture for unmanned systems reference architecture, version 3.2: JAUS Working Group. Retrieved from http://www.jauswg.org/ on February 24, 2007.

Kyriakopoulas, K., & Skounakis, N. (2003, September). Moving obstacle detection for a skid-steered vehicle endowed with a single 2-D laser scanner. IEEE International Conference on Robotics and Automation, 1, 7-12.

Kelly, A., & Stentz, A. (1998, May). Rough terrain autonomous mobility-part 2: an active vision, predictive control approach. Autonomous Robots, No. 5, 163-198.

Kweon, I. (1991). Modeling rugged terrain by mobile robots with multiple sensors. PhD Dissertation, Carnegie Mellon University.

Langer, D., Rosenblatt, J., & Hebert, M. (1994, December). A behavior-based system for off-road navigation. IEEE Journal of Robotics and Automation, 10(6), 776-782.

Matthies, L., & Grandjean, P. (1994, December). Stochastic performance modeling and evaluation of obstacle detectability with imaging range sensors. IEEE Transactions on Robotics and Automation, 10(6), 783-792.

Matthies, L., & Rankin, A. (2003, October). Negative obstacle detection by thermal signature. IEEE International Conference on Intelligent Robots and Systems, 1, 906-913.

Miller, I., Lupashin, S., Zych, N., Moran, P., Schimpf, B., Nathan, A., & Garcia, E. (2006, August). Cornell university's 2005 DARPA Grand Challenge entry. Journal of Field Robotics, 23(8), 625-652.

Murphy, R. (1998, April). Dempster-Shafer theory for sensor fusion in autonomous mobile robots. IEEE transactions on Robotics and Automation, 14(2), 197-206.

Novick, D. (2002, May). Implementation of a sensor fusion based object detection component for an autonomous outdoor vehicle. PhD Dissertation, University of Florida.

Okui, T. (1999). Development of a multi-layered map management system utilizing the nonhomogeneous Markov chain approach. PhD Dissertation, University of Florida.

Royer, E., Bom, J., Dhome, M., Thuilot, B., Lhuillier, M., & Marmoiton, F. (2005, August). Outdoor autonomous navigation using monocular vision. IEEE International Conference on Intelligent Robots and Systems, 1253-1258.

Shafer, G. (1976). A mathematical theory of evidence. Princeton, N.J.

Stentz, A., Kelly, A., Rander, P., Herman, H., Amidi, O., Mandelbaum, R., Salgian, G., & Pedersen, J. (2003, July). Real-time, multi-perspective perception for unmanned ground vehicles, Proceedings of AUVSI2003.

Solanki, S. (2003). Implementation of laser range sensor and dielectric laser mirrors for 3D scanning of glove box environment. Master's Thesis, University of Florida.

Talukder, A., Manduchi, R., Rankin, A., & Matthies, L. (2002, June). Fast and reliable obstacle detection and segmentation for cross-country navigation. IEEE Intelligent Vehicle Symposium, 2, 610-618.

Takeda, N., Wantabe, M., & Kazunori, O. (1996, November). Moving obstacle detection using residual error of FOE estimation. IEEE International Conference and Intelligent Robots and Systems, Osaka, Japan, 3, 1642-1647.

Talukder, A., Manduchi, R., Castano, R., Owens, K., Matthies, L., Castano, A., & Hogg, R. (2002, October). Autonomous terrain characterization and modeling for dynamic control of unmanned vehicles. IEEE Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 1, 708-713.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. The MIT Press, Cambridge, Massachusetts, London, England.

Thrun, S., Burgard, W., & Fox, D. (2000, April). A Real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. IEEE Conference on Robotics and Automation, 1, 321-328.

Trepagnier, P., Nagel, J., Kinney, P., Koutsougeras, C., & Dooner, M. (2006, August). KAT-5: Robust systems for autonomous vehicle navigation in challenging and unknown terrain, Journal of Field Robotics, 23(8), 509-526.

Urmson, C., Ragusa, C., Ray, D., Anhalt, J., Bartz, D., Galatali, T., Gutierrez, A., Johnston, J., Harbaugh, S., Kato, H., Messner, W., Miller, N., Peterson, K., Smith, B., Snider, J., Spiker, S., Ziglar, J., Whittaker, W., Clark, M., Koon, P., Mosher, M., & Struble, J. (2006, August). A robust approach to high-speed navigation for unrehearsed desert terrain. Journal of Field Robotics, 23(8), 467-508.

Wolf, D., Sukhatme, G., Fox, D., & Burgard, W. (2005, April). Autonomous terrain mapping and classification using hidden Markov models. IEEE Conference on Robotics and Automation, 2026-2031.

Wu, H., Siegel, M., Stiefelhagen, R., & Yang, J. (2002, May). Sensor fusion using Dempster Shafer theory, Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference, 1, 7-12.

Xu, J., Wang, H., Guzman, J., Ng, T., Shen, J., & Chan, C. (2003). Isodisparity profile processing for real-time 3D obstacle detection. Proceedings of the IEEE 6th International Conference on Intelligent Transportation Systems, Shanghai, China, 1, 288- 292.

Ye, C., & Borenstein, J. (2003, April). A new terrain mapping method for mobile robots obstacle negotiation. Proceedings of the Unmanned Ground Vehicle Technology Conference at the 2003 SPIE Aerospace Symposium, 5083, 52-62.

Ye, C., & Borenstein, J. (2004, April). T-transformation a new traversability analysis method for terrain navigation. Proceedings of the Unmanned Ground Vehicle Technology Conference at the 2004 SPIE Defense and Security Symposium, Orlando, FL, 473-483.

BIOGRAPHICAL SKETCH

Sanjay Solanki was born in India. He completed his bachelor’s degree in mechanical engineering at Vishwakarma Institute of Technology, Pune. After his graduation he worked as a research and development engineer in Mahindra & Mahindra Ltd., an automobile company in India. After two years of industry experience, Sanjay decided to go back to school. He came to the United States in 2001, to pursue his master’s degree in mechanical engineering at the University of Florida. For the next six years, the Center for Intelligent Machines and Robotics at the University was his home, where he first completed his master’s degree in 2003 and then joined the PhD program. Sanjay has worked on various robotics projects under the guidance of his advisor, Dr Carl Crane. Among the most prestigious projects, Sanjay was actively involved in the development of autonomous vehicles for the DARPA Grand Challenge competition in the year 2004 and 2005. To increase his skills in the robotics field, he added dual minors in computer science and electrical engineering. In May 2007, he received his doctoral degree. He plans to continue his work in the robotic industry.